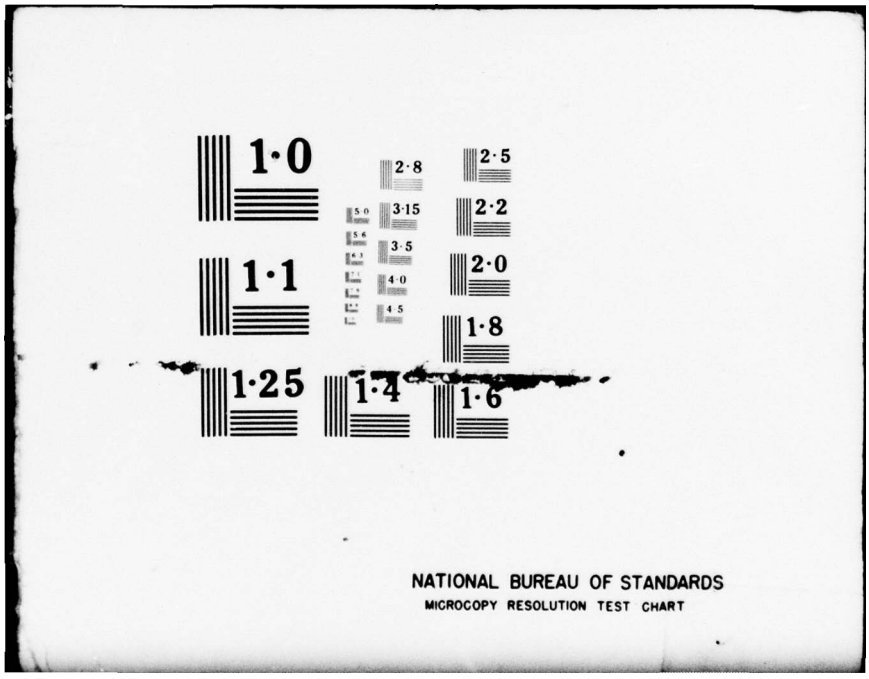
AD-A056 391 DEVELCO INC SUNNYVALE CA F/G 9/2
MAGNETIC BUBBLE MEMORY SYSTEM CONCEPTS FOR FIELD DATA ACQUISITI--ETC(U)
SEP 77 L R BULDUC, T C MOORE, O L KWONG DNA001-77-C-0137
UNCLASSIFIED 1026-77930 DNA-4499F NL

1 OF 2
ADA
056391


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
5·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

AD-E300 265

(12)

LEVEL II

DNA 4499F

AD A056391

MAGNETIC BUBBLE MEMORY SYSTEM CONCEPTS FOR FIELD DATA ACQUISITION APPLICATIONS

Develco, Inc.
404 Tasman Drive
Sunnyvale, California 94086

30 September 1977

DDC
RECEIVED
JUL 20 1978
B

Final Report

DDC FILE COPY

CONTRACT No. DNA 001-77-C-0137

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

THIS WORK SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER RDT&E RMSS CODE B344077462 J11AAXSX35249 H2590D.

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, D. C. 20305

78 06 16 025

Destroy this report when it is no longer needed. Do not return to sender.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: DNA 4499F	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): MAGNETIC BUBBLE MEMORY SYSTEM CONCEPTS FOR FIELD DATA ACQUISITION APPLICATIONS.	5. TYPE OF REPORT & PERIOD COVERED: Final Report.
6. PERFORMING ORG. REPORT NUMBER: 1026-77930	
7. AUTHOR(s): L. R. Bulduc, E. P. Hutchins, T. C. Moore, O. L. Kwong	8. CONTRACT OR GRANT NUMBER(s): DNA 001-77-C-0137
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Develco, Inc., 404 Tasman Drive, Sunnyvale, California 94086	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NWET Subtask J11AAXSX352-49
11. CONTROLLING OFFICE NAME AND ADDRESS: Director, Defense Nuclear Agency, Washington, D.C. 20305	12. REPORT DATE: 30 September 1977
	13. NUMBER OF PAGES: 162
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*	15. SECURITY CLASS *(of this report)*: UNCLASSIFIED
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

12 156 p.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

16 DNA SBIE 19 4499F, AD-E300 265

18. SUPPLEMENTARY NOTES

This work sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B344077462 J11AAXSX35249 H2590D.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Magnetic Bubble Memories
Data Recording
Data Storage
Data Acquisition

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The characteristics of the new solid state, nonvolatile Magnetic Bubble Memories (MBM) were investigated to define their potential for field data acquisition applications. A prototype model of a memory subsystem (hardware and software) was designed and tested to demonstrate one concept. Other memory and acquisition system design alternatives were also considered.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

78 06 16 025

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

410 754 act

SUMMARY

The relatively new solid state, nonvolatile Magnetic Bubble Memory (MBM) devices have excellent potential in field data recording applications. This program was, therefore, undertaken to develop and evaluate a prototype subsystem model in order to assess the applicability in using MBM's for DNA event recording, and make system design recommendations that could be implemented when MBM production versions become available in the near future. Although many companies are engaged in MBM development, only three (Rockwell International*, Texas Instruments, Plessy Memories) have started selling prototype versions at this time. The detail hardware and software design for the prototype memory evaluation system was based on the Texas Instrument version because of its availability and, more importantly, their approach seems to be the most flexible for general applications. A comprehensive test program was developed and used to evaluate the memory subsystem with favorable results. It is our conclusion that instrumentation can and should be developed to take advantage of the unique benefits possible with MBM devices, especially in rigorous field applications.

ACCESSION for
NTIS White Section ☑
DDC Buff Section ☐
UNANNOUNCED ☐
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

*Reference to a company or product name does not imply approval or recommendation of the product by the U.S. Government or by DNA to the exclusion of others that may be suitable.

PREFACE

This work was sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B3440 77462 J11AAX5X352-49H2590D, and was performed under contract number DNA 001-77-C-0137.

We would like to thank the various manufacturers of Magnetic Bubble Memory devices, that we contacted, for their cooperation and assistance. In addition, special thanks is due to Mr. Gerald Cox, of the Texas Instruments MBM Product Marketing group, for his aid on design and evaluation test questions.

CONTENTS

	PAGE
PREFACE	2
1. INTRODUCTION	5
1.1 GENERAL	5
1.2 MBM CHARACTERISTICS	7
1.3 CURRENT MBM ALTERNATIVES	8
1.4 SYSTEM DESIGN CONSIDERATIONS	11
1.5 FUTURE POTENTIAL	14
2. PROTOTYPE MEMORY SYSTEM DESIGN	15
2.1 SYSTEM DESCRIPTION	15
2.2 DETAILED HARDWARE DESCRIPTION	18
2.2.1 Controller Board	18
2.2.2 Branch Control Board	21
2.2.3 FIFO Board	24
2.2.4 Function Timing Board	26
2.2.5 I/O Interface Board	29
2.2.6 Bubble Memory Board	31
2.3 DETAILED SOFTWARE DESCRIPTION	32
2.3.1 PDP-11 Software	32
2.3.2 Bubble Controller Micro Programs	39
2.3.3 Magnetic Bubble Maps	47
2.3.4 Timing Control Program	49
3. TEST RESULTS	50
3.1 OBJECTIVES	50
3.2 BACKGROUND	50
3.3 ERROR RATE TEST PROCEDURE	57
3.4 DATA ANALYSIS	58
3.5 T.I. REEVALUATION OF BUBBLE MEMORIES	61
4. DATA ACQUISITION SYSTEM DESIGN RECOMMENDATIONS	67
4.1 DATA ACQUISITION SYSTEM USING 8 BIT PARALLEL PROTOTYPE MEMORY	69
4.2 ALTERNATE 8 BIT PARALLEL DIGITAL DATA ACQUISITION DESIGN CONCEPT	72

PAGE

4.3 CONTROL CONCEPT FOR A SERIAL ARCHITECTURE MAGNETIC BUBBLE MEMORY 74

REFERENCES 78

APPENDICES

I - Texas Instruments Equipment Specification TBM0101 Bubble Memory Device 79

II - MC2901 Four-Bit Microprocessor Slice 99

III - 9408 Microprogram Sequencer 105

IV - Magnetic Bubble Fortran Test Program 111

V - MBM Subroutines, Macro 121

VI - Program examples 125

VII - MBM Controller Instruction Set (Partial) 131

VIII - Magnetic Bubble Memory Program 135

IX - Example of Texas Instruments MBM Test Data 141

X - MBM Map for 1702 PROM 143

XI - PROM Code for Function Timing Generator 145

XII - Test Report Memos to Texas Instruments (Memo from L. Bulduc, 5 August 1977 and 16 August 1977) 147

SECTION 1 - INTRODUCTION

1.1 GENERAL

Magnetic Bubble Memories (MBM) are solid state devices in which information is stored in the form of a small cylindrical magnetic domain, or bubble, of opposite polarity to the thin layer of magnetic material in which it is located. These devices were first developed at Bell Telephone Laboratories in the 1960's by A.H. Bobeck and others[1], and there are now numerous companies actively engaged in the development of bubble memories for applications ranging from mass memory to digital recording systems.[2,3] Two companies, Rockwell International and Texas Instruments, have just started to sell prototype versions within the last year or so, and one, Plessy Memories, plans to offer sample quantities by the end of 1977. Other organizations engaged in MBM development (e.g.; Bell, IBM, Hewlett-Packard, Intel, other semiconductor manufacturers, and various foreign sources) are either doing so for their own use, or have not made any decision to market them, or have not reached the prototype production stage.

Because of their non-volatile storage characteristics, relatively low power consumption and solid state construction, MBM's have excellent potential in field data recording applications. Therefore, this technology could have a significant impact on the Defense Nuclear Agency's (and others) field experiment programs by offering a way to achieve a reliable, secure, compact, and cost efficient alternative to tape recorders and recording oscilloscopes in many applications. The work described in this report was, therefore, undertaken to develop and evaluate a prototype memory subsystem model in order to make specific recommendations on data acquisition applications and system designs that could be implemented when MBM production versions become available in the near future.

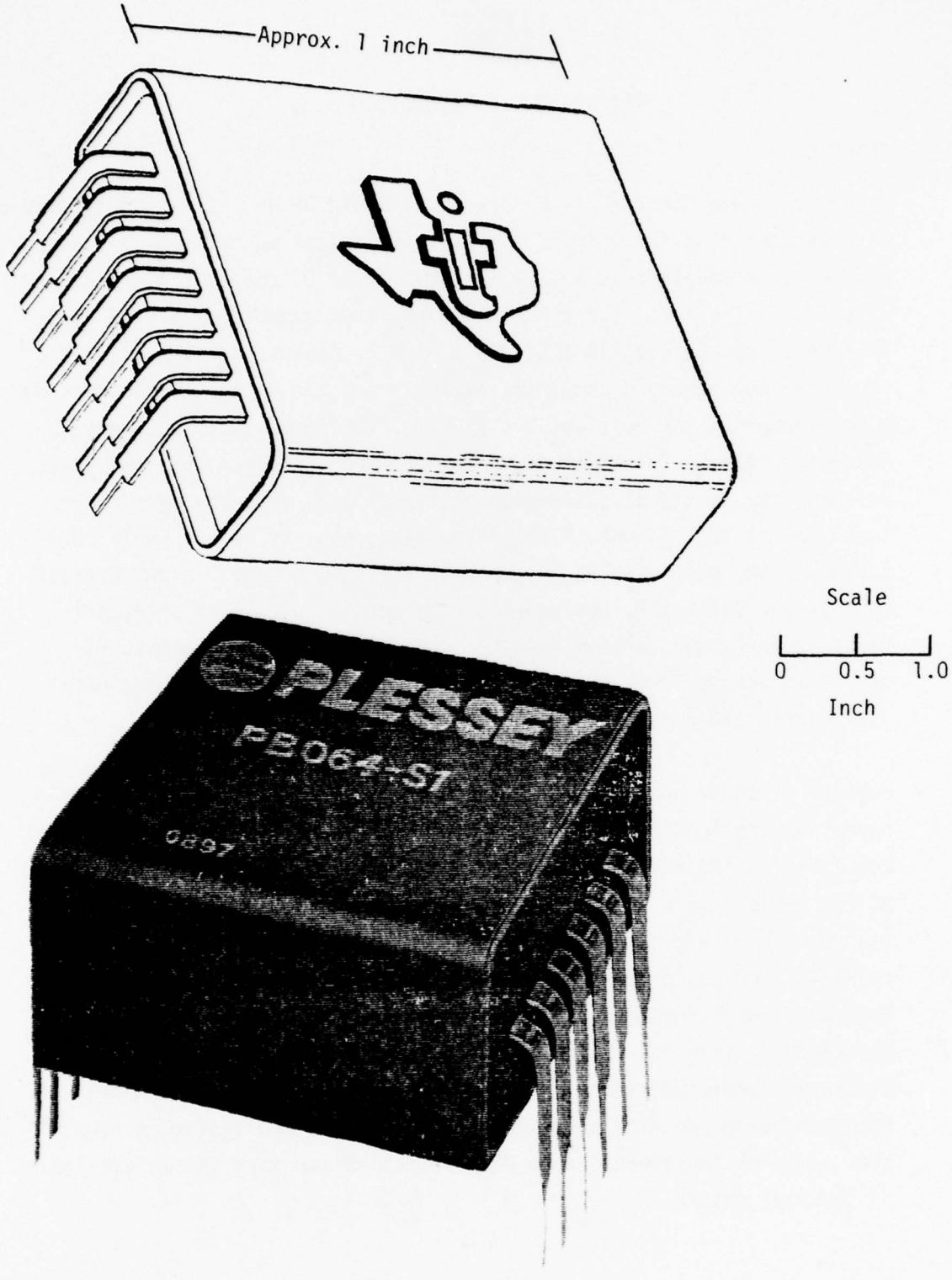


Figure 1-1. Magnetic Bubble Memories

1.2 MBM CHARACTERISTICS

MBM's consist of a wafer, or chip, on which the magnetic bubble circuits are constructed, and external magnets and coils which provide bias, and a rotating magnetic field to move the bubbles. The resulting component, Figure 1-1, is very compact and much like a standard Integrated Circuit (IC) so is inherently quite rugged if the coils are suitably encapsulated. All devices have a self-contained magnetic shield to protect them from external fields less than about 20 to 40 Oersteds, which is roughly 50 to 100 times the earth's field. Peripheral circuits are required for bubble generation, annihilation, control, and sense, as well as to generate rotating magnetic fields. A family of Large Scale Integration (LSI) devices is evolving from several manufacturers for direct MBM interface. There are many ways to organize a MBM chip ranging from a simple endless loop shift register to many parallel shift registers, such as T.I.'s 92K bit major/minor loop arrangement, and others. In addition, there may be one or many chips located within a bias magnet and coil set. Each configuration has significant advantages and disadvantages so it is likely there will be considerable evolution in the industry before some "standard" is settled on. (The theory and operating details of MBM's are extensively described in the literature; see the references and Appendix 1.)

Present manufacturing technology limits the capacity of practical individual chip sizes to about 100k bits, but this is expected to grow rapidly. Thus, 200k bit chips will be available soon, Megabit sizes are forecast for the not too distant future, and 10M bits should be achieved by 1982. Operating speeds of present devices are primarily limited by the resistance and inductance in the coil drive circuits which constrains most practical devices to operate below about 200kHz to 400kHz. The components now on the market all operate at a basic rate of 100kHz, but T.I.'s device is limited to a "data rate" of less than 50kHz by a "branching" operation. However, the mobility of even the relatively large bubbles in present devices is much better than that, and ways may yet be found to operate at speeds of up to 1MHz.

It is projected that eventually self biasing versions will be developed which will be capable of much higher speeds (optimistic forecasts suggest 10's of MHz).

One key advantage of some MBM's is that they can start and stop on a bit so power can be removed between operations in long term recording. However, some manufacturer's devices do not permit storage in the major loop which partially defeats this advantage.

Current devices are typically specified over an operating temperature range of 0°C to 70°C although this span can be readily shifted so a -30°C to +40°C range is covered, e.g., which would be more useful for field applications. The magnetic materials used will probably limit high temperature operations to a little over 100°C. But, operating ranges of -50°C to +100°C have been achieved in the laboratory, and it is reasonable to expect production versions with similar performance can be achieved in the future. The devices are also potentially capable of withstanding fairly severe mechanical shock if appropriate packaging methods are used. T.I. currently has a nonoperating specification of 300g, and has done very limited testing to 3000g. There does not seem to be any reason why MBM's could not also be operated at these levels.

1.3 CURRENT MBM ALTERNATIVES

The Rockwell International (Autonetics Group) development work has emphasized digital recorder subsystems. Their first commercial version, the POS/8, is organized as a 100k byte (8 bit word) endless loop recorder (Simple Single Shift Register), achieved by using 8 chips within a common magnetic bias and drive assembly, that can operate at a maximum rate of 100k bytes/second. In addition, they are developing a 10^8 bit recorder (achieved by large chip arrays) which can operate at a 1MHz to 2MHz rate (achieved by multiplexing) for NASA.[3] Although all memory functions are self-contained, external digitizing, multiplexing,

and control circuitry is required for use in a data acquisition system. The POS/8 is moderately expensive (around $5,000 in small quantities), but it clearly will find use in some field applications, especially those that can use its capabilities directly, because of its current availability and relative simplicity. However, it is unlikely that the simple serial shift register will be implemented in future higher density chips so growth potential, except through the series addition of chips, is limited. Also, the "hybrid" chip packaging approach has limited flexibility for designing integrated general purpose, low cost data acquisition systems.

Texas Instruments and Plessy Memories, on the other hand, have emphasized the single chip approach that is similar to conventional semiconductor memories. That is, each chip has its own bias magnet, drive coils, and external shield in a relatively small dual-in-line type Integrated Circuit (IC) package. This allows considerable flexibility in system design which has obvious advantages in using the devices for diverse applications. There may be some long term price advantages too since this form is more likely to be mass produced, and to have a family of LSI interface devices. There is an efficiency penalty in that more space and support circuitry is required, but the advantages outweigh the problems in many applications.

A more significant problem in data acquisition applications is the fact that the primary market for the single chip devices is as a replacement or alternative to disc and other mass memory forms where random access time is important. This means that some form of major/minor loop architecture, which is difficult to use in straight high speed recording, will predominate until the projected radically different high speed, high density future generation versions emerge from the labs of Bell and IBM. This is further complicated, in the T.I. 92k bit multiloop device at least, by the fact that 13 out of the 157 minor loops are allowed to be bad to compensate for manufacturing yield problems.

(Even when the yield is improved on current low density devices, the push to higher density will keep the problem around for awhile.) It is necessary to keep track of and bypass these bad loops in addition to the normal minor loop transfers, etc. Thus, in straight recording applications a significant amount of additional control circuitry is required to deal with the multiloop structure. However, in advanced data acquisition systems there will be microprocessors, used for automatic data processing and formatting, etc., which can also be used for recording control so the penalty may not be serious.

It is worth noting that Plessy's first device will be a 64k bit serial shift register type that will be useful in simpler applications even though it is of lower capacity. However, they will follow this with a 64k bit multiloop organized device, and it is not clear whether they would continue the simpler serial version for long.

Since the objective of this program was to investigate utilizing MBM's for integrated general purpose low cost data acquisition systems, we chose to base our development work on the single chip devices. It is likely that the discrete devices will be most useful in the long run because of their inherent flexibility, e.g.; many applications can get by with a serial rather than 8 wide organization. Further, the Rockwell/NASA work has amply demonstrated the advantages and drawbacks of the simple shift register architecture in a multiple chip package.[3,5] Because the Texas Instrument device was the only "discrete" device available to us at the start of this program, prototype versions were purchased for use in the prototype memory model. (The Plessy devices will not be available in sample quantities until late in the fourth quarter of this year.) Also, having to work with the more complex T.I. major/minor loop organization (complicated by the bad loop problem) now is really an advantage since it not only represents a worst case design problem, but this, or similar forms, is probably the organization other manufacturers will emphasize as a way to minimize access time,

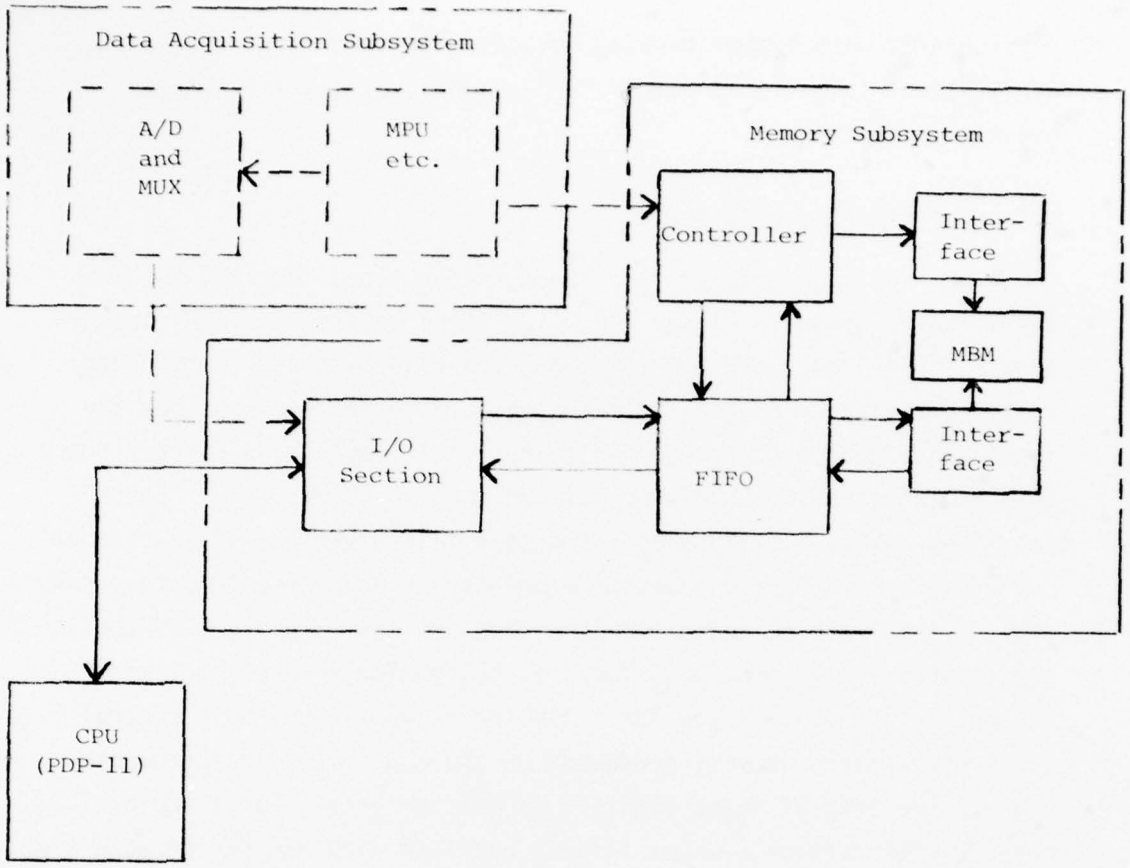
particularly with higher density devices, for their most important applications (e.g.; disc memory replacement), at least in first generation devices.

1.4 SYSTEM DESIGN CONSIDERATIONS

There are two major classes of interface functions required for MBM operation as shown in Figure 1-2. The logic control section formats data (if required), sets up the timing, and keeps track of the read/write operations. A direct MBM chip interface section provides the input drive and output sense functions. T.I. supplies a circuit board that emulates the functions of coil driver, function driver, sense amplifier, etc., specifically for prototype evaluation. This circuit board was used in the evaluation breadboard since these functions should soon be available as dedicated IC's, and are not terribly important to the general system problem. The T.I. design for a function timing generator circuit was used for a similar reason. The logic control section, however, is very dependent on the application requirements, and was the area of major emphasis on this program. For example, T.I. makes a CMOS controller, but it only operates with one device at a time, and cannot be used directly in 8 bit parallel operations. Since high speed 8 bit (or more) parallel operation represents one important category of data acquisition applications, and is the most complex with the T.I. architecture, the prototype memory controller was designed for this case. The design of the controller for the prototype MBM model is governed by two major factors:

a. Sophisticated control should be incorporated in advanced data acquisition systems to handle such functions as automatic calibration, gain setting, adaptive data sampling, and such housekeeping functions as address, event timing, etc., data.

b. Sophisticated control is the easiest way to handle the complex operational requirements of the MBM itself such as the major/minor loop transfers, skipping failed loops, etc.



1) Provides input for test operations
2) Provides data dump for all operations

Figure 1-2. MBM System for data acquisition applications.

By sophisticated control, we basically mean one that has the power and flexibility of a Microprocessor Unit (MPU), or similar device. In order to consider the worst case design problem, a controller that is directly involved in the data handling was chosen, although this may not be the most generally useful way to do it. (See Section 4.2). Since this requires an operating speed greater than that of the Motorola 6800 MPU, for example, the Advanced Micro Devices AM2901 four bit slice bipolar cascadable microprocessor was chosen to achieve the speed and computational power required. This achieved, in effect, a "nano-processor" with considerable add-on capability. It is also needed to interface with more complex functions, e.g.; the MPU in a complete data acquisition system. A further advantage of this approach was that it permitted getting the memory system up and running with a minimum amount of design and software effort. Even so, the resulting design is fairly powerful on its own. For example, not only can it perform the complex MBM control functions for a simple fixed sample rate acquisition in its present form, it could be made to do responsive data acquisition with additional software, or to do limited adaptive data acquisition, event time, and address recording with some additional hardware and software.

A First-In/First-Out (FIFO) buffer memory was also used on the prototype system to provide the required "elasticity" to deal with the bad minor loop problem characteristic of the T.I. "prototype" MBM devices. Note that this function can simply be eliminated when T.I. achieves 100% device yield as eventually expected in "production" devices. An input/output (I/O) section is used to interface with a PDP-11 minicomputer which serves to provide selected data pattern inputs for test operations, and to dump data stored in the memory for all operations. The computer also is very useful for analyzing the data, and as an aid in trouble shooting for test work. For use in a data acquisition system, an A/D converter and multiplexer function would simply be inserted in place of the computer for acquisition, and disconnected for dump. (The details of the system design and test results are discussed extensively in the following sections.)

1.5 FUTURE POTENTIAL

The long term technical potential for MBM devices appears to be very good. They are capable of being operated under fairly extreme environmental conditions, and appear destined to fill a gap in the speed/cost spectrum between existing forms of solid state memories and large mass memory systems. The advantages of non-volatility are obvious regardless of whether the application is cash registers or unattended data recording. MBM technology is radically different, and in its infancy, so it is reasonable to expect that current technical problems will be overcome in the near future. However, the ultimate impact MBM's will have on the market depends to a very great extend on whether they can be made cost competitive. For example, it is estimated that 100k byte memory subsystems, based on production versions of today's devices, could be sold for around $1,500, or about 0.2 cents per bit, which is higher than most electro-mechanical methods. But, forecasts show that 10M bit chips, and prices per bit of 0.01 cents are possible by 1982, which is competitive with most other methods; so the overall prospects do appear to be good. Therefore, it is our conclusion that MBM technology will survive, in a competitive sense, and that instrumentation should be developed to take advantage of the unique cost and performance benefits that are possible with MBM devices, especially in rigorous field applications.

SECTION 2 - PROTOTYPE MEMORY SYSTEM DESIGN

2.1 SYSTEM DESCRIPTION

The prototype Magnetic Bubble Memory system consists of both hardware and software. The hardware consists of 8 MBM boards and 5 control boards as shown in the block diagram, Figure 2-1. The software consists of micro programs for the MBM controller and PDP-11 programs for communications with the MBM controller. These items are discussed in more detail in the following sections.

A Digital Equipment Corporation PDP-11/05 Computer is used as a means of communicating with the MBM controller for tests and data dump operations. It instructs the controller to read or write a particular file from the MBM. Data is transmitted to or from the memory through 8 First In-First Out (FIFO) devices. Also, on the FIFO board is a Programmable Read Only Memory (PROM) which contains information about the "bad" loops in the MBMs. This map prevents the generation of data in bad loops or the reading of data from bad loops. Commands from the PDP-11 are received by the controller through a branch control board which causes the controller to branch to a Read or Write routine. The PDP-11 sends the first word containing the command and the two most significant bits of the file address. When the controller receives the first word it sends an indication to the PDP-11 which causes the remaining 8 bits of the file address to be transmitted to the controller. In the case of a write-into-memory routine, the PDP-11 sends data to the FIFOs until a FIFO Full signal is received from the controller. It waits until the FIFO is not full and then sends more data. This process is repeated until 144 bytes of information are transferred to the FIFOs.

In the case of a read-from-memory the PDP-11 waits until a FIFO not empty signal is received from the controller and then reads data from

the FIFO until the FIFO EMPTY signal is received. The PDP-11 then waits until it has received 144 bytes of information.

The MBM controller board sends signals to the PDP-11 and other function controls via 2 latches on the branch control board. After the controller receives the instruction and file address it enables the MBM boards and shifts the bubbles in the minor loops the number of times specified by the file address. A Transmit Out enable signal is sent to the function timing board which transmits the file (1 bit from each minor loop) into the major loop. The bubbles in the loops are then shifted a number of times (68) then the controller either enables the replicate (read) or annihilate (write) signals from the timing board. The controller shifts the bubbles either 19 times in the read mode or 228 times in the write mode before enabling the FIFO clock and the Detect Strobe (read) or Generate Signal (write) from the timing board. These signals are enabled every other cycle (data occurs every other position) of the bubbles until 157 bits (including zeros in the bad loop positions) are transferred. The bubbles are then shifted until they are again over the minor loops and then transferred in. The bubbles are then shifted until they are back in the same position in the minor loop as they were prior to the start of the operation. Note that data storage in the major loop is not permissible with current T.I. devices. The controller then inhibits or resets all necessary devices and waits for another instruction.

The Magnetic Bubble Memory prototype system is contained in a card file with 14 card positions as follows:

(5) Controller Boards
(8) MBM Boards
(1) Spare

Each board plugs into a 44 pin wire wrap connector on the back plane where power inputs and most interboard signals are wired. However, some inter-controller board communication is accomplished through ribbon cables.

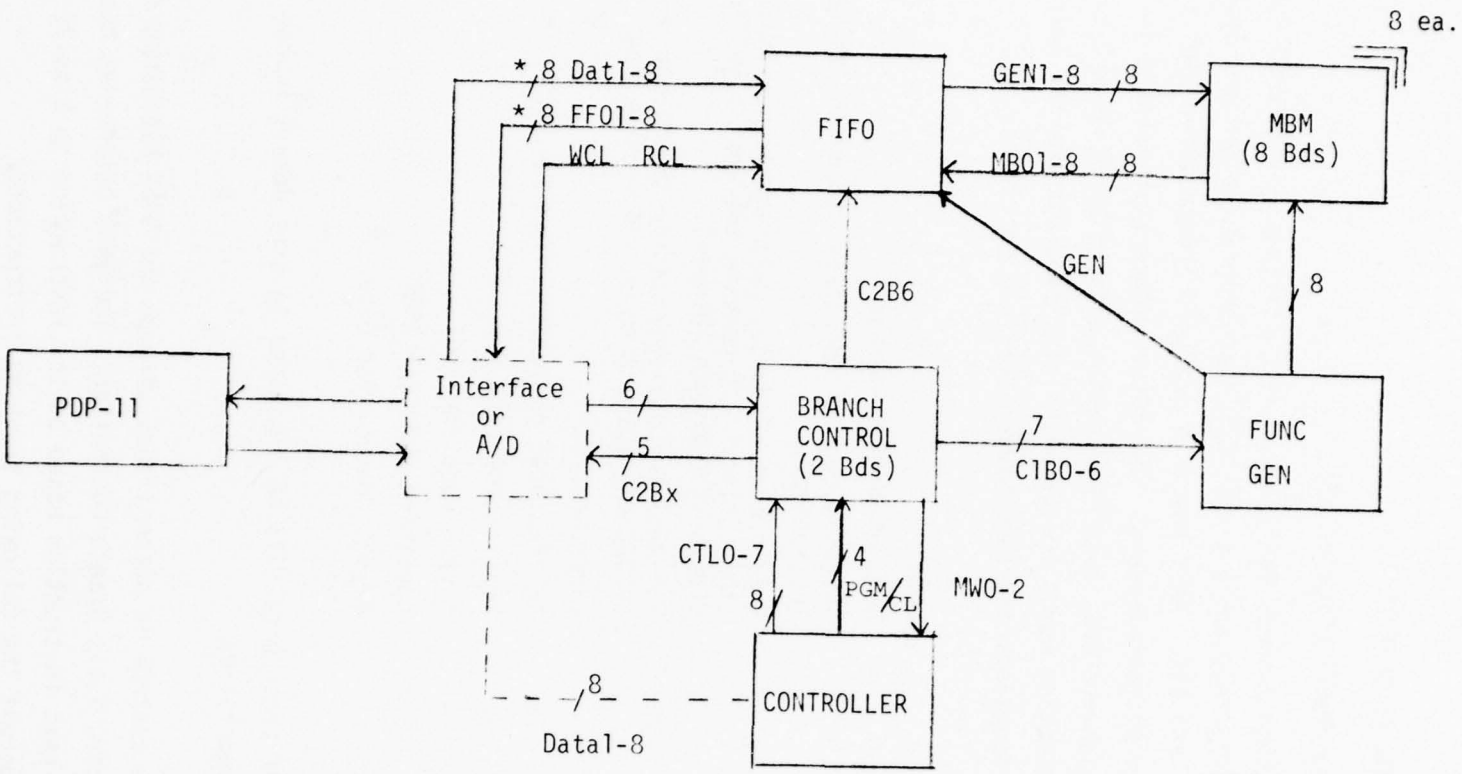


Figure 2-1. Block Diagram MBM System

2.2 DETAILED HARDWARE DESCRIPTION

1. Controller Board (Figure 2-2)

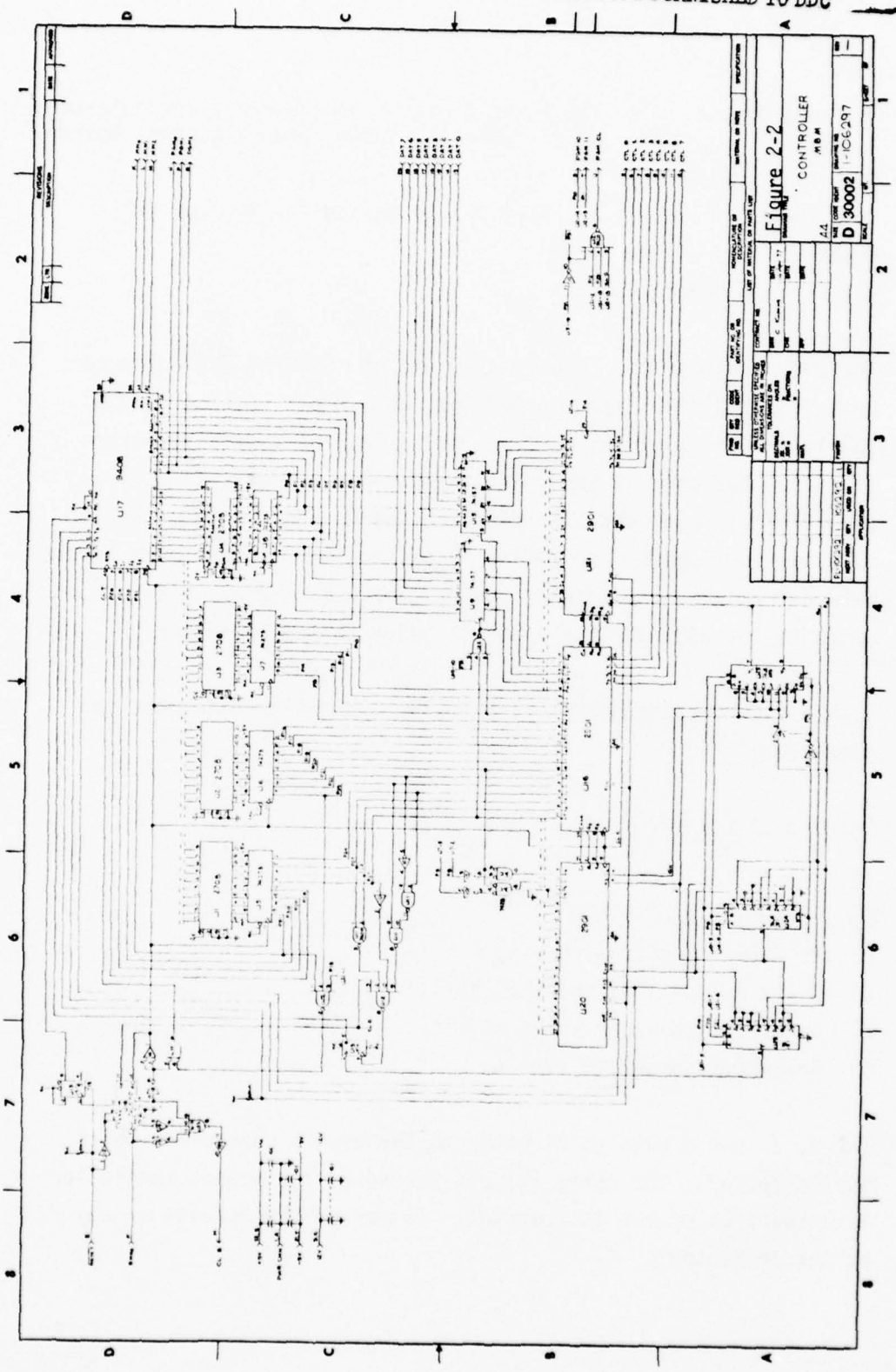
This section assumes familiarity with workings of the 2901 Four Bit Slice Microprocessor and the 9408 Microprogram Sequencer, Appendices II and III. This board is designed around three 2901 Four Bit Slice Microprocessors. The micro programs are stored in four 2708 Erasable PROMS, each PROM having a capacity of 1k x 8 bits. The instruction words are 32 bits long, some bits having specific control functions and some having multiple functions.

BITS 0 - 7	Data
BITS 0 - 9	Data of Branch Address
BITS 8 - 9	Shift control of the carry bit during shift instructions.
BITS 0 - 2	Branch control during Multiway Branch instructions (bits 3 - 9 contain the most significant bits of the Branch Address).
BITS 10 - 13	"A" Register designation for the Microprocessor or I/O control during an I/O instruction.
BIT 14	Carry Control
BITS 15 - 18	"B" Register designation for the 2901
BITS 19 - 21	Destination code for 2901
BITS 22 - 24	Source code for 2901
BITS 25 - 27	Operation code for 2901
BITS 28 - 31	Branch Control for 9408

Usage of these bits will be discussed in more detail in the software description.

Program control is accomplished through the 9408 Microprogram Sequencer, i.e.; Branch Controller. The 9408 determines the address of the next instruction based on the instruction in bits 28 - 31 which permit the following types of instruction:

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC



Multiway Branch - The lower 3 bits of the address are determined by an external source (branch control board).

Unconditional Branch

Conditional Branch - Any 1 of 4 bits tested for True or False.

Branch to Subroutine

Return from Subroutine

The 9408 controls the 10 address lines of the 2708 PROM directly. Data from the PROMS are clocked into four 8 bit latches (74273). The lower 8 bits of the word go to 2 quad 1 of 2 data selectors (74157). The other 8 inputs to the 74157's are data lines from the PDP-11 through the I/O interface board. Selection of which data are received by the 2901's is a function of other bits in the word determining the type of instruction. Bits 8 and 9 are also gated to the 2901 to load a 10 bit value to the processor.

Eighteen other bits go directly to the 2901 which determine the operation.

Outputs of the three 2901's are as follows:

a) 8 bits for control
b) Results = Ø (Z bit)
c) Results = Negative (N bit)
d) Carry out on left shift (C bit)
e) Carry out on right shift (C bit)
f) Arithmetic overflow (V bit)

The V, Z, and N bits go directly to the branch controller which may be tested. The carry (C) bit tested by the branch controller is a function of the instruction. It may be the results of any of the following:

1) C_n+4 of the most significant 2901
2) R_3 of the most significant 2901
3) Q_3 of the most significant 2901
4) Q_0 of the least significant 2901
5) R_0 of the least significant 2901

The R_3, Q_3, R_0, and Q_0 inputs and outputs are also a function of the word such that arithmetic shifts or rotates can be accomplished.

Along with the 8 control line outputs two address lines PGM10 and PGM11 and a clock are generated for steering and clocking data on the branch control board.

Also generated on the controller board are the 4 clocks required for operation (See Timing Diagram, Figure 2-3).

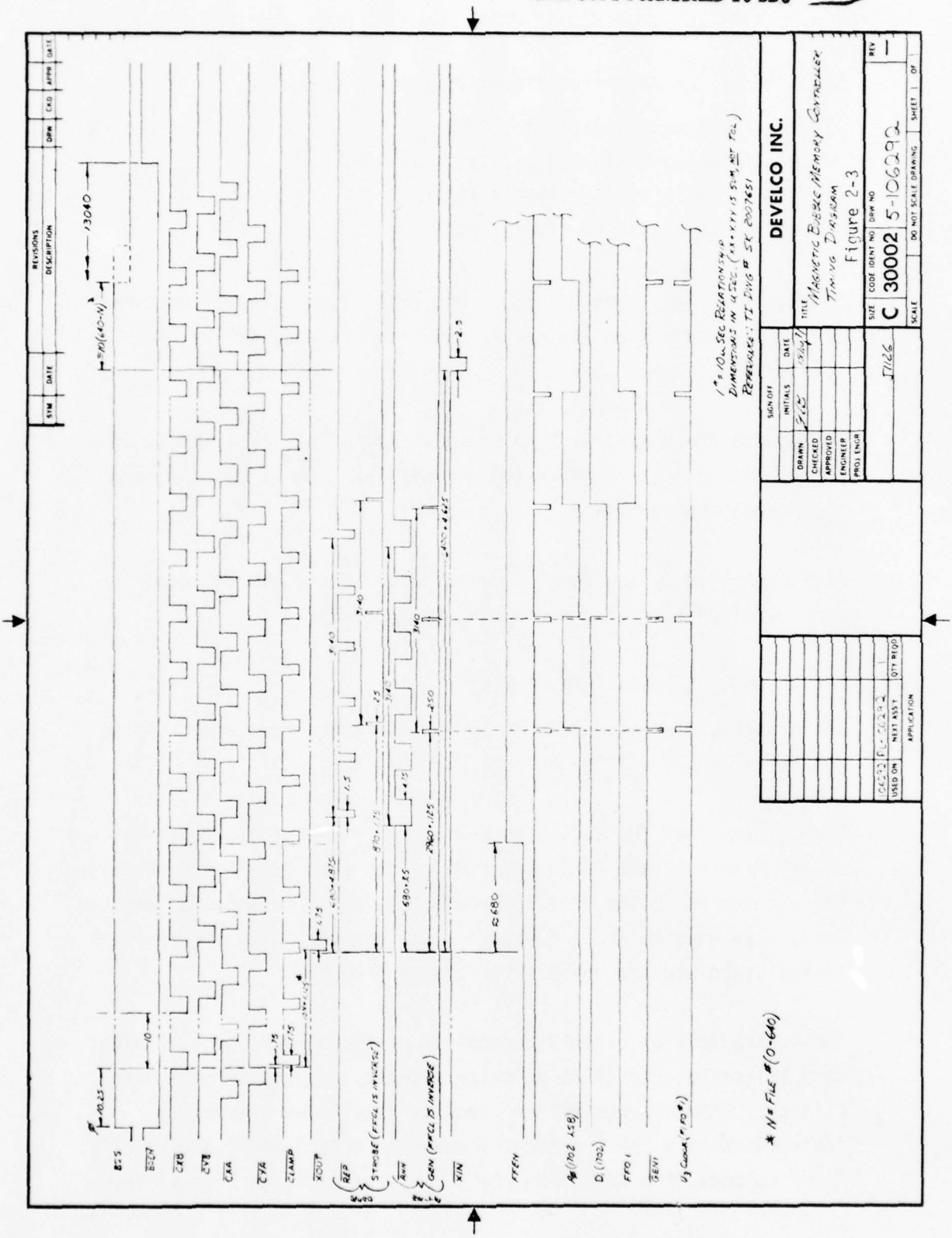
2. Branch Control Board (Figure 2-4)

The branch control consists of 2 output latches for data from the controller board and branch control circuitry.

Output data from the 2901's is clocked into 1 of 2 eight bit latches as determined by the PGM10 and PGM11 lines also from the controller. C2B outputs go to the PDP-11 through the I/O board and C2B6 (Write/Read) also goes to the FIFO board. C1B outputs go to the function timing board and are enables for timing signals.

The branch control circuitry consists of three 74151, one of eight data selectors, one 74148 priority encoder, and one 74L5175 Quad D latch. When the controller executes a multiway branch, instruction bits 0 - 2 determine the gate selection of the 74151's. The 3 outputs then determine the 3 least significant bits of the

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC



THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

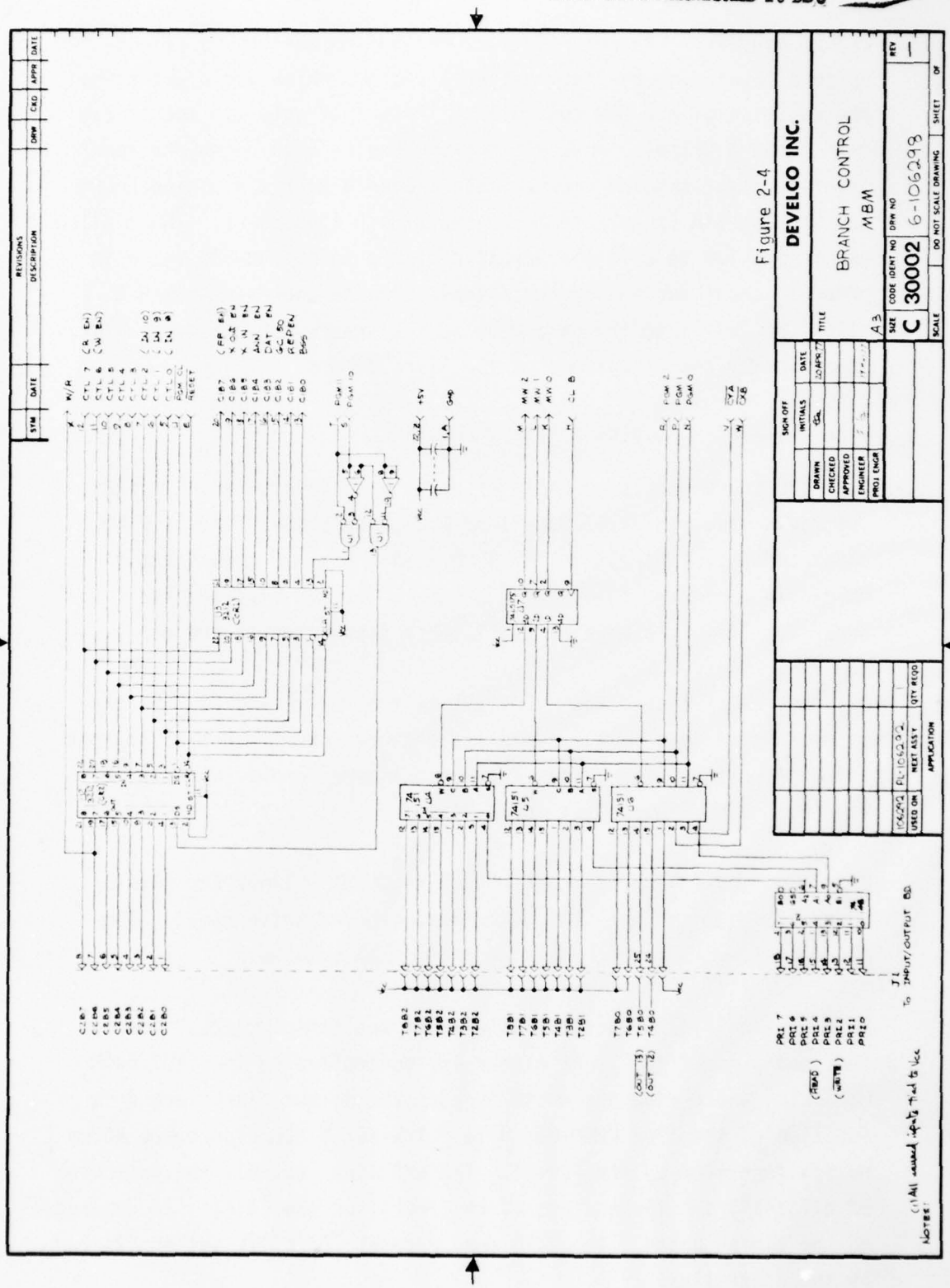


Figure 2-4

branch address. The priority encoder (branch address 0) allows up to 8 input commands from external sources which would determine the function of the MBM controller. Presently only two inputs are used (Read & Write). However, another device with higher or lower priority could command another task. Branch addresses 1 and 2 are the CXB and CYA signals from the function timing board. These allow the controller to wait for specific points in the bubble cycle in order to count and set control lines. Branch addresses 4 and 5 allow the PDP-11 to indicate when data is ready for the controller, specifically the 2 portions of the file address.

3. FIFO Board (Figure 2-5)

The FIFO board mainly consists of eight 9403 64-bit First In/First Out registers, six 74157 Quad 1 of 2 data selectors for the FIFO clock and data control, a 1702 PROM (256 x 8 bits) containing the bubble map, and two 74161 4-bit counters for the 1702 address control. (Refer to Section 2.3.3 for a discussion of the map.)

Data are clocked into the FIFOS via control of U1 and U2. In the case of write-into-memory, data are entered from the PDP-11 through the DATØ-7 lines. In a read from memory process data are clocked in from the MBM boards via the MBOØ-7 lines.

Data are clocked out of the FIFOS to the FFOØ-7 lines (to PDP-11 in read mode), and the GENØ-GEN7 gates (in the write mode). The GEN signal comes from the timing board when enabled.

Four of the 74157's gate the FIFO input and output clocks. During the read process the input clocks are controlled by the 1702 PROM Map data, and during the write process the output clocks are a function of the 1702 PROM Map data. The RCL & WCL clocks are 400ms pulses generated by the PDP-11. The W/R line controls the switching of all 74157's. The address clock (FFCL) for the 74161's is generated on the timing board. The FIFO enable/reset (CRFF) is set or cleared by the controller.

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

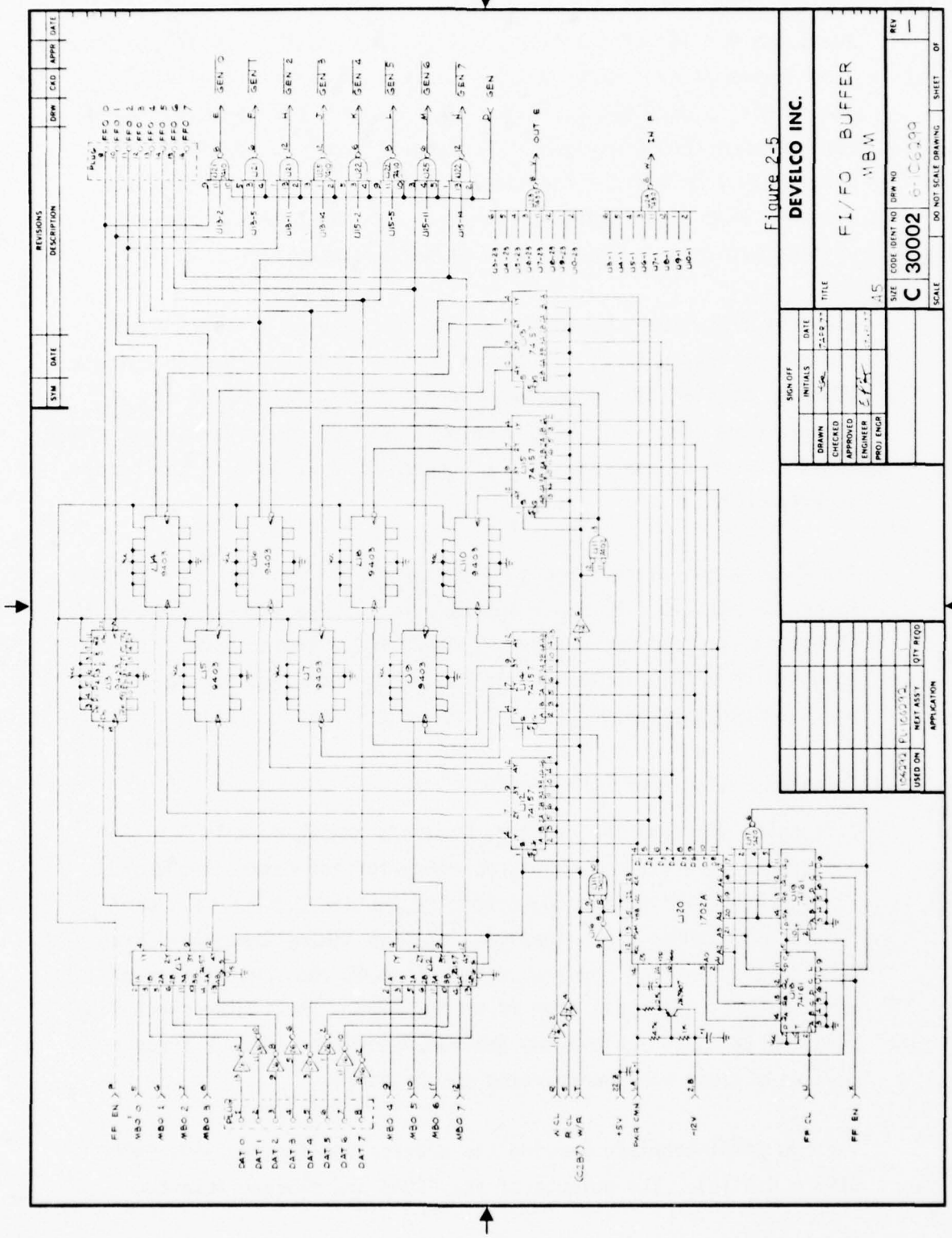


Figure 2-5

Addresses 0 - 157 of the 1702 are programmed with a map of the minor loop status of each MBM. Of the 157 locations there are 144 1's and 13 Ø's in each bit position. The Ø's indicate bad minor loops. For instance WORD Ø may consist of minor loop Ø on boards 1 - 6, minor loop 1 on board 7 and minor loop 2 on board 8. The output from the PROM also enables or inhibits the GEN gates to prevent writing 1's into bad loops which is not allowed.

When the FFCL line goes high, the address counter is clocked and the appropriate pair of 74157's is ungated causing any high outputs produced by the 1702 to go low which will clock the FIFOS if outputs were previously high. This in essence says the data at the input is good or the data at the output was written into the bubble, so advance the FIFO.

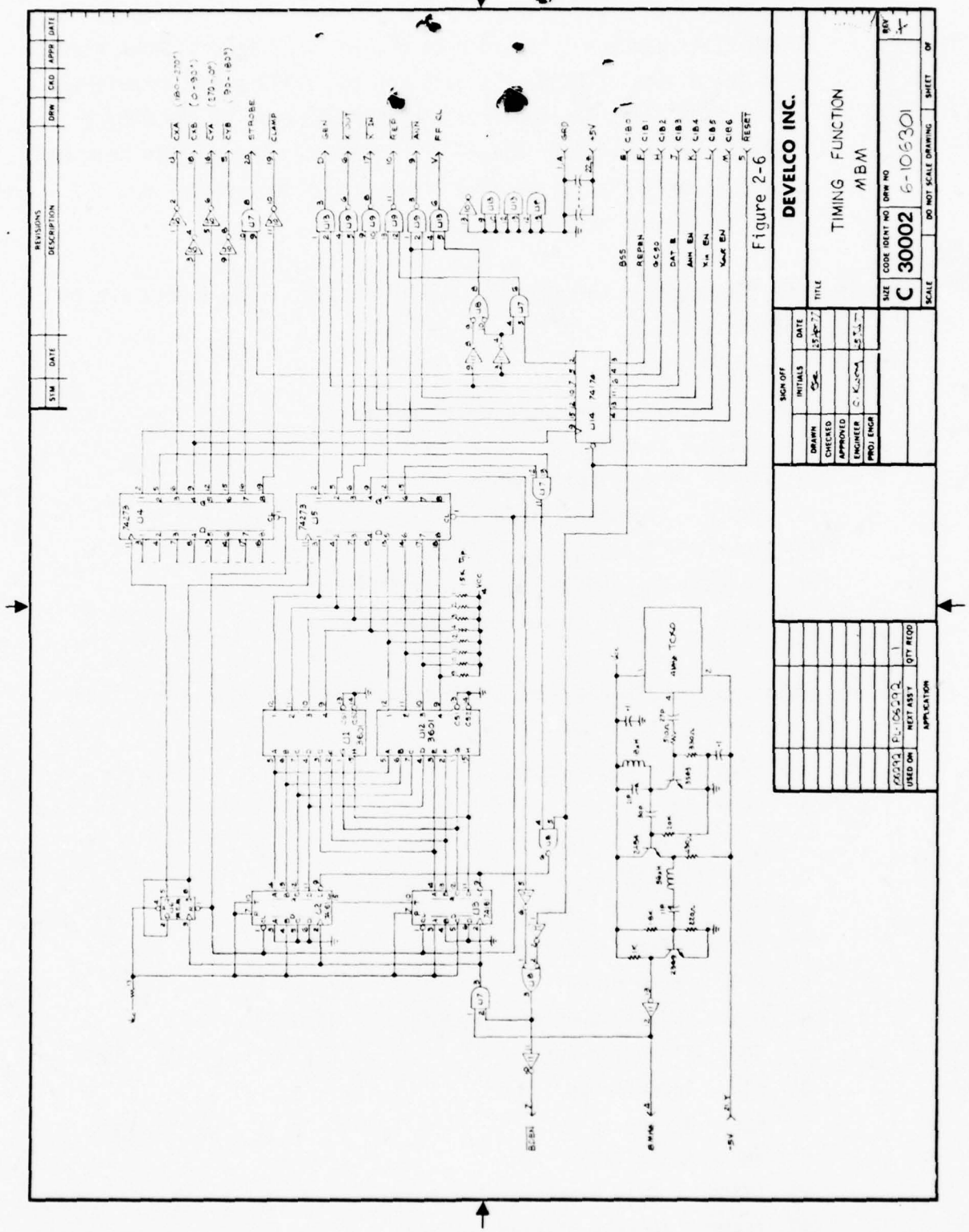
Two FIFO status signals are sent to the PDP-11 via the I/O boards. These signals are FIFO Empty and FIFO Full. The lines are true if the status is true for any one of the FIFOS. These signals are used by the PDP-11 in loading or unloading the FIFOS during the Write and Read processes.

4. Function Timing Board (Figure 2-6)

This board produces the required accurate timing signals for the MBM control, FIFO clock, and 8MHz clock for the controller board. The function timing code, described in Section 2.3.4, is stored in two 3601 PROMS. Note that the Function Timing Generator Circuit is basically the T.I. design adapted to suit the requirements of this system. The oscillator is a 4MHz TCXO. The timing circuit requires 8MHz, to achieve the desired resolution, thus a frequency doubler circuit was incorporated on the board.

The two 74161 Counters provide the address for the two 3601 PROMS (256 x 4 bits). The outputs of the PROMS are clocked into two

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC



74273 8-bit latches. In order to obtain 14 different clock signals from the 8 bits of PROM, the data for the 74273's are interleaved and each 74273 is clocked from a different phase of the 4MHz 2 phase clock produced by the 7474. Therefore, the outputs from U4 and U5 are relative to each other in .125µs increments, and all signal durations are in multiples of .25µs.

The timing board has 8 inputs and 14 outputs. The inputs are as follows:

1) $\overline{\text{Reset}}$
2) BSS - Enables timing and the MBM board
3) REPEN - Enables replicate
4) ANNEN - Annihilate Enable
5) XINEN - Transfer in Enable
6) XOUTEN - Transfer out Enable
7) DATEN - Enables GEN output and FIFO clock
8) GC50 - Enables Strobe and FIFO clock

Inputs 3 - 8 (above) are clocked into a 74174 6-bit latch. The outputs are as follows:

A) BDEN - MBM board Enable
B) 8MHz clock
C) $\overline{\text{CXB}}$ - 0-90° of the coil drive
D) $\overline{\text{CYB}}$ - 90-180° of the coil drive
E) $\overline{\text{CXA}}$ - 180-270° of the coil drive
F) $\overline{\text{CYA}}$ - 270-0° of the coil drive
G) FFCL - FIFO clock
H) GEN - Enables Generate gates on FIFO board
I) $\overline{\text{XOUT}}$ - Transfer Out
J) $\overline{\text{XIN}}$ - Transfer In
K) $\overline{\text{REP}}$ - Replicate
L) $\overline{\text{ANN}}$ - Annihilate
M) STROBE - Data clock for MBM board
N) CLAMP - Sense amplifier control

5. I/O Interface Board (Figure 2-7)

The I/O (Input/Output) interface board interfaces the PDP-11 to the controller boards. The PDP-11 interfaces to the controller through a Digital Equipment Corporation DR11-C general device interface.

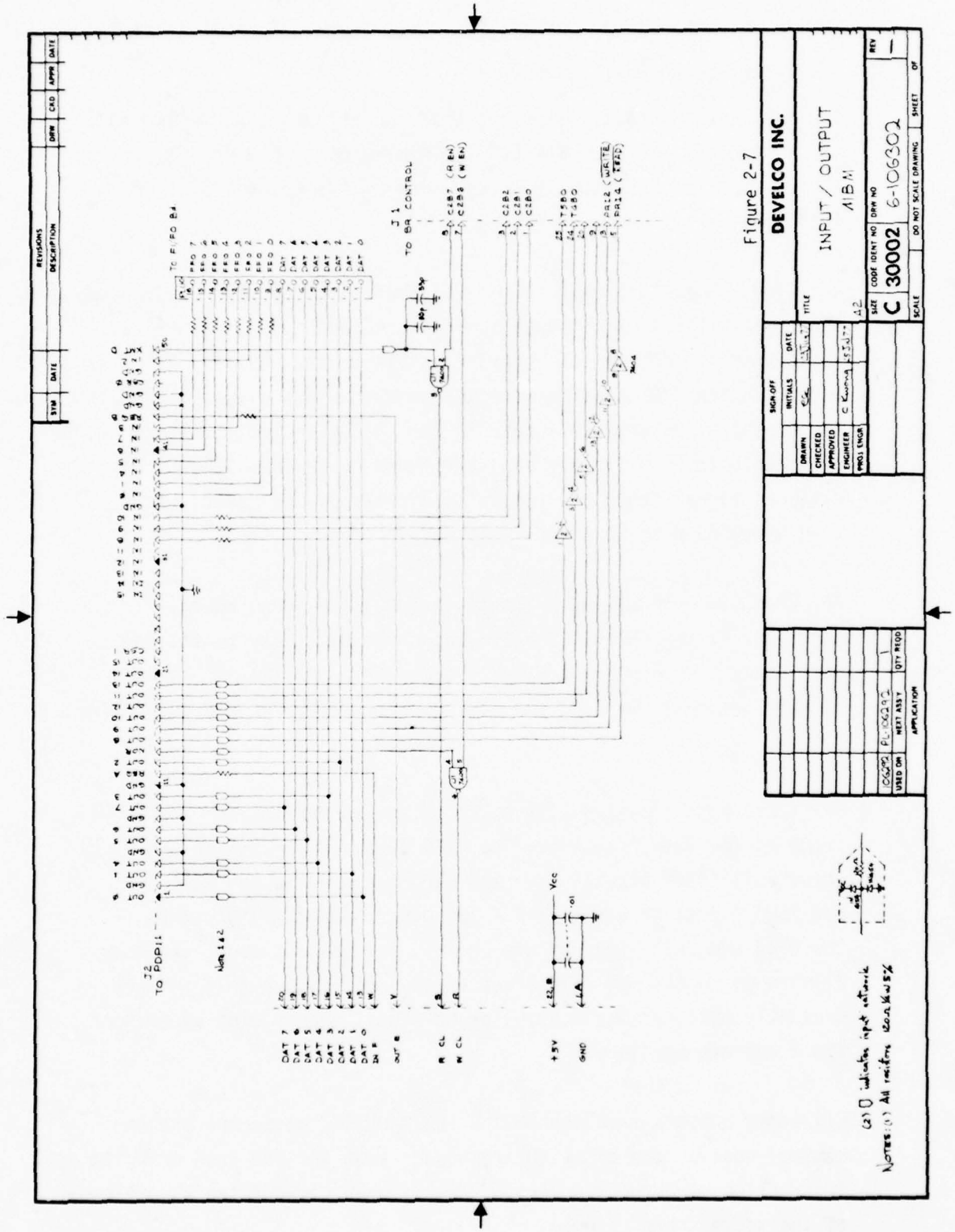
All lines on the I/O board are terminated with 120 ohm resistors to reduce the rise time of signals, and thus reduce cross coupling noise over the 25 feet of interface cable to the PDP-11. All inputs to the I/O board are terminated with diode clamps to Ø volts and +5 volts, as are the inputs to the DR11-C to limit voltage transients to .6 volts within Ø volts and +5 volts. Other individual signal functions were slowed even more by capacitors if they were found to be excessively noisy.

The Read Clock (RCL) and Write Clock (WCL) are 400ns pulses generated by the PDP-11, and correspond respectively to the New Data Ready and Data Transmitted signals from the DR11-C. These signals are gated by C2B6 and C2B7 controlled by the MBM controller board.

The least significant 8 bits of input and output to/from the PDP-11 are used for Data Transfer. The FIFO Output Empty (OUTE) and FIFO Input Full (INF) signals from the FIFO board go to the REQA, and REQB inputs to the DR11-C. Two other inputs are wired to the CSRØ and CRS1 lines of the DR11-C, but are not used. Another 3 lines go to the IN8-IN10 lines of the DR11-C, but only two are presently used for the controller to signal the receipt of each of the 2 address portions.

Six other outputs from the DR11-C (OUT8-OUT14) go to the branch control board, four of which are used. (Two for the Read or Write Instruction, and two for branch control when each of the 2 portions of the address are ready.)

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC



6. Bubble Memory Board (Figure 2-8)

(Refer to the Timing Diagram, Figure 2-6, and the current MBM Specification, Appendix I.)

The MBM Boards (8) were designed by Texas Instruments specifically for prototype evaluation purposes. The board may be discussed in three parts, i.e.; bubble shifting control, bubble function (generate, transfer, etc.) controls, and sensing.

The current in the X and Y coils controls the shifting of the bubbles in the TBM0101. The currents in these coils are controlled by the $\overline{CXB}$, $\overline{CYB}$, $\overline{CSA}$, and $\overline{CYA}$ input signals. The alternating currents in the coils are produced by two 75325 Field Drivers and diode matrices at their outputs.

Individual control of bubbles is accomplished by driving three gates with a specified pulse width and current amplitude. The three control gates are Transfer, Replicate/Annihilate, and Generate. The five board input signals are $\overline{XIN}$, $\overline{XOUT}$, $\overline{REP}$, $\overline{ANN}$, and $\overline{GEN}$ each received by 1/2 of a 75463 Dual Peripheral Driver. The output of each 1/2 of the 75463 controls a constant current source determining the amplitude of the signals to the three control gates. The width and timing relationship of the control signals are determined by the function timing board.

During the Read process, the output of the MBM is detected by a MC1444 sense amplifier. The output of the sense amplifier is clocked into 1/2 7474 as determined by the strobe input from the timing board. The output of the 7474 is buffered through 1/2 of the 75463 peripheral driver. The $\overline{Clamp}$ input from the timing board to the MC1444 enables the input to be detected when high. During the Write process the $\overline{Gen}$ line being low will cause a bubble to be created. During the Read process the presence of a bubble will produce a low output.

2.3 DETAILED SOFTWARE DESCRIPTION

This section will discuss in detail four items:

1) PDP-11 Programs
2) MBM controller micro programs
3) MBM map programming
4) *Timing control program*

Items 1 and 2 are interrelated in that each item relies, in part, on the other to produce specific signals as a result of the programming, i.e.; it is a handshaking process. For sections 2.3.1 and 2.3.2 it is assumed that the reader has a basic knowledge or understanding of software. However, it may be helpful to refer to the Digital Equipment Corporation handbooks, references 7 and 8.

2.3.1 PDP-11 Software

General Description - The programs for testing MBM's and communicating with the MBM controller were written in two computer languages, 1) Fortran, and 2) PDP-11 Assembly language. The reason for this is that direct communications with the MBM controller requires the setting and clearing of bits which is more easily accomplished in Assembly language than in Fortran. However, Fortran facilitates easier analysis of data and implementation of test programs.

A. Fortran Program (Appendix IV)

The Fortran program has four main functions which may be selected at the start of the program.

1) Write - Writes 144 ASCII characters from the keyboard to a specified file in memory.
2) Read - Reads 144 ASCII characters from a specified file in memory and outputs them on the teletype.
3) Dump - Performs the same function as Read, but for all files from the specified start to end.

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

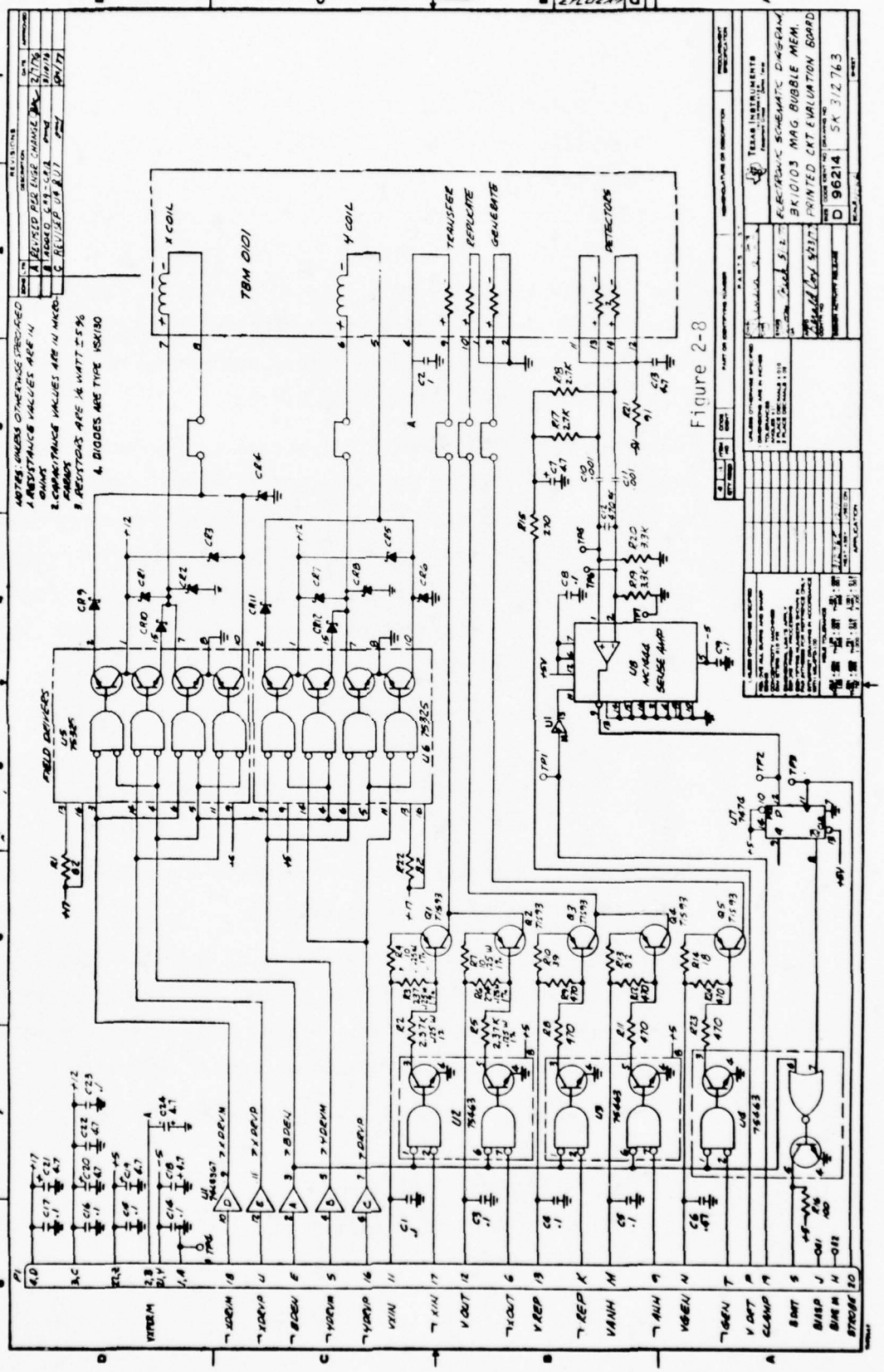


Figure 2-8

4) Test - Has 3 modes that may be requested as follows: (Refer to Section 3 for additional information on the use of these tests.)

a. Specify Pattern - Allows a pattern of up to 10 byte values to be specified. This pattern will be written into the file (5) specified and read back the specified number of times. This process can also be repeated a specified number of times on each file before proceeding to the next file (if more than 1 file is specified).

b. File Integrity Test - This test writes a file number based pattern into all files and then all files are read in succession. The number of write passes and number of read passes/write pass are specified by the operator. On every other write pass the file address based pattern is compli-mented, e.g.; on the first and every other write pass the pattern is as follows:

High byte of address
Low byte of address
Ones compliment of high byte of address
Ones compliment of low byte of address

On the second and every other write pass the pattern is the one's compliment of the above pattern.

c. Generated Pattern Test - Combination of both the Specify Pattern and File Integrity Test. Eight different patterns are written into the memories. Each pattern being repeated every 8 write passes. The patterns are as follows:

1) Word 0 = 0, Word 1 = -1, Word 2 = 0, etc.
2) Compliment of 1
3) All words = -86
4) Compliment of 3 (All = 85)
5) All words = 0

6) All words = -1
7) File Integrity Pattern
8) Compliment of 7

NOTE: Values are 2's compliment decimal value of binary pattern, e.g.; -1 = all 1's, etc.

The above patterns were designed to test combinations of all or alternating bubble patterns in major and minor loops and to evaluate system noise conditions.

Detailed description of Fortran software is as follows:

Three single dimension byte arrays are defined as:

A) Data to/from MBM (145 bytes)
B) Data from keyboard
C) Pattern array

The ASCII character definitions are expected inputs from the keyboard to the various questions as follows:

Lines 100 - 901 ask for and analyze the answer to the question of what type of operation, (Read, Write, Test, or Dump). Single character responses are sufficient.

Lines 200 - 812 perform the single file ASCII read routine and write to keyboard.

Lines 300 - 820 perform the single file ASCII write routine getting the data from the keyboard.

Lines 600 - 601 perform the Dump routine (multiple file ASCII read).

If the answer to the programs first question is "T" (test) the program will jump to line 2000 where the question will be asked; What type of test? The answer may be "S" (Specify Pattern), "F" (File Integrity), or "G" (Generate Pattern).

In any case, all three routines use common sections of the program. In order to decrease the run time and increase test throughput subroutines were not used. Instead, two control words were used for program control - KNTRL and KNTRLØ.

The common section of the code used by all three test routines is between 714 and 762. If it is an "S" type test both control words = Ø. If it is an "F" type test, KNTRLØ = Ø. and KNTRL = +1 or -1. If it is a "G" type test. KNTRL = Ø, and KNTRLØ = 1 - 6 for the first six patterns. For patterns 7 and 8 KNTRL = -1 or +1 and KNTRLØ = 7.

When a "S" type test is requested the program will go to 700 where it begins to ask and analyze the answers to a series of questions which determine the parameters of the test. These questions are:

- The first and last file numbers to test.
- Number of byte patterns and their values.
- Number of errors to print.
- Number of Write and Reads/Write Pass.

By line 720 these questions have been answered.

Lines 720 through 703 print the header for error analysis. The program then enters the common routine. If the type of test is "F" or "G", lines 2009 - 3002 ask the common questions needed for both tests,

- Number of errors to print.
- Number of Write and Reads/Write Passes.

Both tests automatically test all 641 files.

If the type of test is "F" the program will continue at 3002, and set up values required by the common routine to perform the File Integrity test. Lines 3300 to 3320 set up the file number based

pattern and then go to the common routine. Since each file must be written in succession or read in succession the common routine will return to 3200 (Write) or 3600 (Read) based upon the state of KNTRL. After 641 files are written, the program will go to 3500 where conditions are set for the read process.

Lines 4000 - 4201 analyze the results of the test and determine if any more read or write passes are to be executed.

If the type of test is "G", program control goes to line 5000. Lines 5000 - 5006 initialize control values and set up one of six patterns.

Lines 6000 - 6002 set the values required by the common routine, print the test pattern and number on the terminal, and go to the common routine.

The common routine returns to line 9000. Until the first six patterns are generated the program will go to the next pattern generator. Otherwise initial values will be set for generating the 2 file integrity test patterns and program control will then go to line 3005.

The file integrity test returns to line 9900 where any errors are accumulated, KNTRLØ is reset, and control goes to line 9001. This process is repeated until the specified number of write passes is performed.

The common routine starting at 714 sets the A array to the values in the C array modulo NUMPAT (number of patterns), and then calls the macro subroutine (MACWRT), which writes the data into the bubble. MACRED routine is then called, the data being sorted in the A array.

The A array is then compared against the C array modulo NUMPAT looking for errors. If errors are found the board number (NUMBRD) on which the error was found is calculated and then the following information is printed.

Write Pass Number (KNTWRT)
Read Pass Number (KNTRD)
Good Data [C(K)]
Bad Data [A(J)]
Board Number (NUMBRD)
Word Number (NUMWRD)

This process is repeated for all 144 words in the file (NUMWRD), until the specified number of Read (NUMRD) passes and Write (NUMWRT) passes have been executed.

In all of the above test modes the operator may specify the maximum number of errors to be printed to the keyboard before terminating that portion of the test. In such a case the program would advance to the next file or write pass.

B. Assembly Language Programs (Appendix V)

The two subroutines called by the Fortran program are MACRED and MACWRT (Read and Write respectively). They in turn call another subroutine called ADDXMT which performs the task of sending the file address to the controller. MACRED and MACWRT initially set location MASKI with a value which will indicate to the controller to either read or write. They then call ADDXMT.

ADDXMT adds to MASKI the two most significant bits of the address in the two LSB of the 16 bit PDP-11 word. This word is then moved to address 167772_8 which is the DR11-C output register. ADDXMT then readys the second word which will contain the lower

8 bits of the file address along with a branch control bit (020000). ADDXMT then waits in loop 2 until the controller signals it has received the first word before sending the second word. ADDXMT then points to the core address of the data block and returns to MACRED or MACWRT. In each case both will then wait for the controller to indicate it has received the second byte of information and is ready to proceed with the data transfer.

In the case of Write, MACWRT will send the first 60 bytes of data to the eight 64 x 1 FIFO's (Loop 4). It will then go to loop 5 and move another byte of data only when it has an indication that the FIFO is not full. MACWRT will remain in loop 5 until 144 bytes have been transmitted and then return to the Fortran program.

In the case of Read, MACRED will go to loop 7 after it has received a ready indication from the controller. In loop 7 MACRED will remove data from the FIFO's as long as it receives a FIFO not empty indication. It will wait until it has 144 bytes of information before returning to the Fortran program.

2.3.2 MBM Controller Micro Programs

The heart of the MBM controller consists of three 2901 4 Bit Slice Microprocessors. This allows 12 bit data words. However, only 10 bit values may be accessed from PROM, and only 8 bit values are available from the I/O data lines.

The microprocessor and peripheral functions are controlled by a 32 bit word programmed in 2708 PROMS. The purpose of this section is to provide some insight into writing programs for this controller. However, due to the ability of a single word to produce double operand, double source, and double destinations (including I/O) instructions, it would be an extremely lengthy task to document the full capabilities.

A functional description of the 32 bit word is as follows:

BIT NO.

0 - 2 Multiway branch control (will determine bit 0 - 2 of the branch address).

3 - 9 Most significant bits of a Multiway Branch Address.

0 - 9 Absolute Branch Address.

0 - 7 Absolute Data when the data are from PROM and a shift instruction is present (B21 set).

0 - 9 Absolute Data when the data are from PROM and a shift instruction is not present.

8 - 9 Shift control. Will set the input to the LSB or MSB of the Q or R bidirectional ports to a value determined by the above outputs. The bit of the word will be set to that value during a shift instruction. Table 1 below shows the logic for this function.

*10 - 13 "A" Register Address designation if the source code does not indicate the "A" Register address as a source code (B23 = Ø).

10 - 13 I/O Control if (B23 = 1) in which case the bits are as follows:

B10, B11 Output Register Address
B12 Output Designation
B13 Input Designation

Note: I/O instructions require that the Branch Control Instruction (B28-B31) not be a conditional branch instruction (e.g.; $<8_H$).

*14 Carry bit - significant only when the Op Code B25-B26 is Ø, 1, or 2.

*15 - 18 "B" register address designation

*19 - 21 Destination Code

*22 - 24 Source Code

*25 - 27 Op Code

**28 - 31 Branch Control

Appendix VII contains a list of some of the more common simple instructions along with a few of the more complicated possibilities. Refer to appendix II which shows the operation source and destination codes for the 2901. However, due to the 2901 and hardware design limitations, caution should be exercised under certain circumstances.

When using source codes 0, 1, 4, and 5 (Register at A address) there may not be a read from the I/O data lines. The data may only be read from PROM.

When loading 10 bits of data from PROM the destination code may not be 4, 5, 6, or 7 since these are shift codes, and B8 and B9 double as shift control bits.

When the instruction consists of a conditional branch, data may not be loaded from the I/O lines. A write into I/O control registers is not allowed 1) On conditional branch instruction, or 2) When "A" Address Register is called for in the source code. Conditional branches do not have the capability of testing more than one bit. Multiple instructions must be written and each except for the last must not effect the test bits.

The carry test bit (charted in Table 2-1) will be set or cleared as a result of the operation if the Op Code is 0, 1, or 2, and the Source Code is 0, 1, 4, 5, or 6. (Otherwise not effected.)

---

*See Appendix II (2901 Instruction Set)

**See Appendix III (9408 Instruction Set)

Table 2-1. Table of carry control and shift control logic

INPUTS		RESULTS				
Destination Code (Octal) (B19-B21)	Shift Control (B8, B9)	C	Q_0	R_0	Q_{11}	R_{11}
0	X	Cn+12	~~X~~	NONE ~~X~~	~~X~~	~~X~~
1	X	Cn+12	~~X~~	NONE ~~X~~	~~X~~	~~X~~
2	X	Cn+12	~~X~~	NONE ~~X~~	~~X~~	~~X~~
3	X	Cn+12	~~X~~	NONE ~~X~~	~~X~~	~~X~~
4 → Q&R	0	Q_0	S	S	R_0	Q_0
	1	Q_0	S	S	R_0	Cn+12
	2	Q_0	S	S	R_0	N
	3	Q_0	S	S	R_0	Ø
5 →	0	R_0	ZS	S	ZR_0	R_0
	1	R_0	ZS	S	ZR_0	Cn+12
	2	R_0	ZS	S	ZR_0	N
	3	R_0	ZS	S	ZR_0	Ø
6 ↔ Q&R	0	R_{11}	R_{11}	Q_{11}	S	S
	1	R_{11}	R_{11}	Q_{11}	S	S
	2	R_{11}	Ø	Q_{11}	S	S
	3	R_{11}	Ø	Q_{11}	S	S
7 ↔	0	R_{11}	Z	R_{11}	Z	S
	1	R_{11}	Z	R_{11}	Z	S
	2	R_{11}	Z	Ø	Z	S
	3	R_{11}	Z	Ø	Z	S

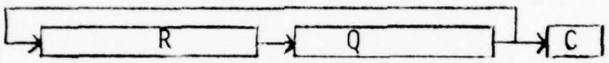
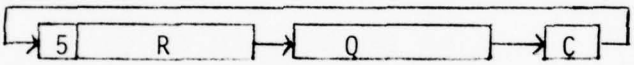
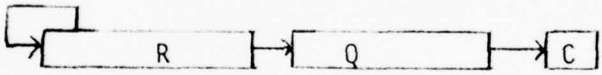
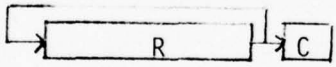
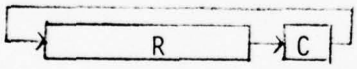
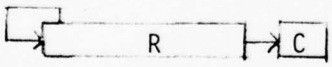
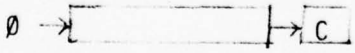
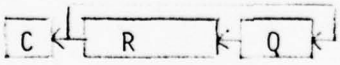
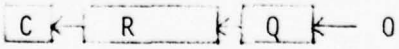
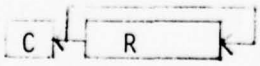
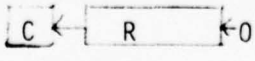
Table 2-1 continued.

S = Signal Source

X = Don't Care

Z = High Impedance State

Therefore, the following shift combinations are available

Dest. Code	Shift Code		MNEMONIC
4	0		DCR
4	1		DRR
4	2		DASR
4	3		DLSR
5	0		CIR
5	1		ROR
5	2		ASR
5	3		LSR
6	0 or 1		DCL
6	2 or 3		DASL
7	0 or 1		CIL
7	2 or 3		ASL

The V (overflow bit) will be reset on the next instruction following the arithmetic operation.

The Z (Results = Ø) and N (Results = minus) bits will be reset on the next instruction, but may be set or reset accordingly by testing the results during any previous condition branch instructions.

Therefore, if V is to be tested following an arithmetic operation, it should be tested first along with a test results instruction. (The Z, V, N, and C bits are a result of the previous instruction.)

Appendix VIII is the listing of the actual program used in the prototype MBM controller. A brief description elaborating on the more complicated instructions follows.

When the hardware is reset the branch controller sets the Program Counter (PC) to Address Ø. The instruction is a multiway branch which will cause the PC to become any of the possibilities between ØØ0 and ØØ7. The results of Op Code 4_8 and Source Code 3_8 is Ø, Appendix II. The results are stored in both REG Ø (Destination Code 3 and B Address Ø) and I/O control REG 1 (Bits 1Ø - 13).

When the PDP-11 sends a Read instruction ($PC \leftarrow ØØ4_H$) or a Write instruction ($PC \leftarrow ØØ2_H$), the instructions at both addresses clears REG 1 and CR2. Both instructions are unconditional branch instructions which will cause the $PC \leftarrow ØØ8_H$ for Write or $Ø8Ø_H$ for Read.

The first instruction at both locations is simply branch to subroutine $Ø5Ø_H$. Though the OP and SRC codes in this instruction $Ø_8$, $Ø_8$ call for A+Q function, the destination code only outputs the results and does not affect the values in any internal registers. Since the I/O control is O_H the results of the operation are not stored.

The subroutine at Ø5Ø receives the file address from the PDP-11, shifts the minor loops a number of times equal to the file address, and determines the number of shifts that will be required at the end of the operation, transfers the bubbles to the major loop, shifts the bubbles until the first bubble is at the Replicate/Annihilate port, and returns to the calling routine.

The instruction at Ø5Ø is a BMW4 (BMW = Branch Multi-Way) instruction which will cause the PC←Ø51 when the 2 MSB's of the file address are present on the data inputs. The bits when received must be multiplied by 2^8. The instruction of Ø51 loads the data, multiplies R15 by 2 an additional 7 times, and also signals the PDP-11 - data received.

Line Ø57 is a No Operation (NOP) instruction which puts the next BMW instruction at a BMW base address (BO-BZ of address are Ø).

The program waits here until the PDP-11 indicates the second portion of the file address is ready, and then PC←Ø59 where the data are stored in R14. Ø5A loads R4 with 641, which is the number of positions in a minor loop. Ø5B adds R14 to R15 resulting in a 1Ø bit file address. If the address is Ø the minor loops are already in a position to be transferred out, and the PC←Ø61. If the address ≠ Ø then the bubbles must be shifted a number of times equal to the file address. Ø5D stores the Value Ø1 in CR1 which enables the timing and MBM boards. Ø5E is a BSRØ28 (BSR = Branch to Subroutine) (described in detail below) which will return after a complete bubble cycle. Ø5F decrements the file address. Ø6Ø decrements R4 which will become the shift count after the bubbles have been transferred in at the end of the operation. Ø6Ø also tests the results of Ø5F, and if ≠ Ø branches to Ø5D. When the minor loops have been shifted as required, line Ø61 adds $4Ø_H$ to CR1 which enables the Transfer Out (XOUT) pulse from the timing board. Ø62 executes a BSRØ28H to wait for 1 cycle completion, and then Ø63 loads R2 with 67., the number of additional shifts

required, until the first bubble position is at the Annihilate/ Replicate port. Ø64 inhibits the XOUT signal. Ø65 - Ø67 shifts the bubbles 67 times. Line Ø68 returns to the next instruction of the calling routine either Ø81 or ØØ9.

In the case of Write, ØØ9 stores 42_H at CR2 signaling the PDP-11 that it has received all information and is ready to proceed, and sets the W/R line to 1. Line ØØA stores a 91_H at CR1 which keeps the Master Enable high, enables the FIFO and the Annihilate gates. Lines ØØB - ØØE shifts the bubbles 228 times annihilating bubbles present in the major loop until the first bubble position is over the Generate port. Lines ØØF to Ø15 go through 157 cycles (314 bubble cycles) of enabling the generation of a bubble in every other position of the major loop while continuing to annihilate any bubbles which appear at the Annihilate port.

Lines Ø16 to Ø1A inhibit the Annihilate and shift the bubbles an additional 3Ø times until they are over the minor loop transfer in ports at which time Ø1B branches to common routine Ø4Ø which will end the operation.

In the case of Read, Subroutine Ø5Ø returns to Ø81 where Ø3 is stored at CR1 enabling Replicate. Line Ø82 - Ø85 shifts the bubbles 19 more times and signals the PDP-11 that all data have been received and the read process started.

Line Ø85 is somewhat confusing in that data to be stored and a conditional branch address are within 1 of each other. The instruction is literally interpreted as follows:

Load Ø83
Decrement by 1 (=$Ø82_H$)
Store in R1 and CR2
Branch if R2 is ≠ Ø to Address Ø83

Lines Ø86 - Ø8D enables the FIFO, keeps the master enable high, and repeating 157 times (314 bubble cycles) enabling the data strobe every other bubble cycle.

Instruction Ø8D loads R2 with 239 which is the number of shifts now required to position the bubbles over the minor loops. Line Ø8E inhibits the Replicate gate. PC←018 where the bubbles will be shifted 239 times before proceeding to 040H to end the operation.

Instruction line Ø4Ø to Ø44 enables the XIN gate for 1 cycle, and shifts the bubbles a number of times until R4 becomes Ø. The value in R4 was determined by subroutine Ø5Ø, and is the number of shift values required to return all bubbles to the same positions they were in before the operation proceeded.

Instructions Ø46 and Ø47 clear CR1 and CR2 and return to instruction ØØØ where the program again waits for a Multiway Branch.

Subroutine Ø28 consists of 2 Multiway Branch Instructions (Ø28 and Ø3Ø) again at addresses where BØ-B2 = Ø.

The BMW1 instruction at Ø28 waits for the 27Ø° signal first to insure that the same Ø° is not counted twice. PC←Ø3Ø where the BMW instruction waits for the Ø° signal and then returns to the calling program.

2.3.3 Magnetic Bubble Memory Maps

When TBM0101 MBM's are received from the Texas Instruments, up to 13 of the 157 minor loops may be marked as unusable. T.I. sends a computer printout of test data with each unit, which also includes their map of the good and bad minor loops.

In the Prototype MBM controller, the maps of the 8 memories used are programmed into a 1702 (256 x 8 bit) PROM. The map for each memory occupies 1 bit position in the PROM.

Appendix IX is a sample of the T.I. computer printout of the MBM test data. The T.I. map is located below the date, and consists of 10 groups of 4 hexadecimal digits with the last "digit" being always a 7 (if good) or F (if bad) since only 157 bits are required. When the Hex map is converted into a binary map, 1's indicate a bad minor loop position. The left or MSB represents minor loop #0 (assuming loops are numbered 0 - 156) and is the one closest to the detector. The value NBIT on the T.I. map indicates the total number of bad loops. Example:

1C04,0000,0000,B600,0000,0000,0000,0000,0000,0007 NBIT 9

0001 110000000100 (Binary Equivalent)

In the above example of the first 16 bit binary value, loops 3, 4, 5, and 13 are bad. (Starting with loop Ø) In the controller, words 0 - 11 would consist of minor loops 0, 1, 2, 6, 7, 8, 9, 10, 11, 12, 14, and 15. Since only 9 loops were bad in this MBM, an additional 4 would be added to make the maps of all memories have 144 loops marked usable. Therefore, the last group of 4 "digits" would become 007F.

In the 1702 map, bad loops are marked by the presence of a 0. Therefore, the T.I. map must be complimented. Appendix X shows a sample System map used to program the PROM. (Rather then complimenting the T.I. map, each position corresponding to bad loops are marked as 0, and all unmarked positions are interpreted as 1.) The 8 bit binary values are then converted to Hex for PROM programming.

2.3.4 Timing Control Program

Appendix XI is a sample of the program in the 2 PROMS on the timing control boards, each PROM having 4 bits of the 8 bit words.

Fourteen controls are contained in the 8 bits. This is accomplished by interleaving the control bits (see Section 2.2.4, function timing board).

Data in even numbered words control 8 functions, and odd number words control 6 functions. A table follows:

Table 2-2. Control Function

Bit #	Even Word	Odd Word
0	Annihilate/FIFO Clock	GEN
1	*PROM Address Counter Load	XOUT
2	CXA	XIN
3	CXB/Enable Latch Clock	BRD EN LATCH
4	CYA	Replicate
5	CYB	*PROM Address Counter Load
6	Strobe	
7	Clamp	

*AND Function

The timing controls from Address ØØ to 51 rotates the current vector in the shift coils from the origin through 27Ø° to Ø°. The timing from 52 to A1 is repeated until the B1BØ line (BSS) is set to Ø by the controller. The instruction from A2 to FE returns the current vector to the origin. FF unlatches the BRD Enable, and disables the address counter clock.

SECTION 3 - TEST RESULTS

3.1 OBJECTIVES

Since bubble memories have many new and unique characteristics (and problems), it was not obvious at the outset, what the test methods and objectives should be. However, as a result of several weeks of intensive effort in learning the problems and characteristics of the device, several specific objectives emerged.

1. Determine and characterize the errors of the prototype system.
2. Determine the nature or source of the errors as being due to
   a. System related factors
   b. Pattern sensitivity
   c. Operating specification marginality
   d. Manufacturer's Quality Control
3. Determine the correct operating parameters and requirements as they relate to the system and the device performance.

It must be emphasized when interpreting the test results, that the tests were conducted on prototype versions of a new technology memory, and the problems encountered will be solved before the devices reach production status. What is important is that a comprenensive test has been developed which can detect failure modes unique to MBM's, and that the tests show present prototype devices, although imperfect, could be successfully used for data acquisition purposes as they are.

3.2 BACKGROUND

The final tests conducted on the system, though the basis for the majority of conclusions in this report, only represented a small portion of the actual energy spent during the evaluation of the system.

The process from the beginning of the evaluation to the point of defining objectives, and conducting the final error rate tests can be

characterized as a "round-robin bootstrapping process." This process consisted of initially exercising the system, through the use of very simple software in the PDP-11, in order to locate controller hardware and software problems. When the basic problems had been overcome the software capability was increased in the PDP-11 which uncovered further errors in the controller software and PDP-11/MBM controller communication noise problems.

As these problems were being solved it became apparent that the final test software in the PDP-11 would have to provide some degree of diagnosis on the errors detected. This allowed characterization of the errors which in turn allowed the possible sources of the errors to be identified. Because of the unique characteristics of MBM's, a number of direct conversations with T.I. engineers were also found to be necessary when interpreting various problems encountered with the memories.

One example is that information received during the design phase led us to believe that writing into "bad" loops was a "don't care" situation. The hardware was designed accordingly. During the evaluation phase it was learned that this was not always so, and bubbles could get trapped in the bad loops and cause problems. The hardware was modified by taking the bubble map outputs and gating them with the data and GEN signals to create the $\overline{\text{GEN}}$ signals for each memory. Also, a procedure was verbally obtained from T.I. to eliminate any bubbles trapped in bad loops, as a result of writing bubbles in them, that might "migrate" or "seed bubbles" causing errors.

Restoration of a device so affected can be accomplished by collapsing the bias field within the memory which maintains the magnetic bubbles. The procedure consists of writing all bubbles into memory. A permanent magnet with a strength of >40/Oe is moved towards the MBM while reading the memory. With the field perpendicular to the top

surface of the memory case, the magnet is slowly moved closer until all bubbles are permanently annihilated. Note, this procedure should be used with caution. If the external field becomes to large, it can reverse the internal field.

The restoration procedure produced only the "relative success" T.I. suggested it would. For example, it was used on 7 units with error rates $>1 \times 10^{-6}$, and produced error rate decreases in all units. However, the error rates of 4 units only decreased by a factor of 2-5, where the remaining 3 units improved considerably, and no errors were found in 2.6×10^{-7} bits.

Measurements on the control currents of the least improved memories indicated that some were not within published operating specifications. Further circuit analysis and conversations with T.I. indicated the need to change the voltage on the control circuits and/or modify (trim) the T.I. control circuit designs. The following changes were necessary:

1. Increase X, Y control voltage to 13.5Vdc. (Required an additional power supply voltage.)
2. Adjust Transfer Out (and in some cases Transfer In) currents for each memory. It was found that the XOUT circuit could not be a constant current source at 12.5Vdc over the range in values of the Transfer coil resistance.
3. Change the current resistor in the Replicate current source to 36 ohms.

It must be pointed out that T.I.'s MBM circuit board is a preliminary engineering design intended for prototype evaluation only, and is not intended to reflect the performance they plan to achieve in production IC versions of the interface functions. Further, there is insufficient statistical data on the MBM's to be able to precisely define nominal

values and tolerances, etc., so these kinds of problems are not unusual at this stage.

The failure of the current control circuits to meet specifications lead to scrutinizing the timing control where a conflict between specifications and the actual requirements were also found. The conflict was in the leading edge timing of the Replicate pulse, generated by the PROM Program (Section 2.3.4), which was found to be 4.875μs instead of the 5.125μs minimum specified in the MBM data sheet. Texas Instruments could not tell us what was optimum for our units, and short term tests indicated no effect so the timing was returned to the programmed value of 4.875μs.

It should be noted here that while the specifications were said to be optimum for the units in our possession, T.I. was respecifying parameters for their latest production units. For some parameters these changes were significant, relative to tolerances. For instance the nominal Generate amplitude had changed from 250mA to 280mA which was previously specified as a maximum. The nominal Replicate and Annihilate leading edges had been changed approximately to what had been the previous max values. The nominal XOUT and XIN leading edges were increased by an amount 2 to 3 times the previous tolerance value. Again, this variation is not unusual at the present stage of development, but it does require a constant awareness of the factors involved.

Also, during this period it was determined that the power up/down sequence was critical. If the correct procedure was not followed it was discovered that current pulses might occur in the memory causing bubble annihilation or generation. A sequence, Table 3-1, was developed which allowed system power control without affecting the data in memory. If an error was made in the procedure, the memory fields were collapsed by the procedure mentioned above to prevent possible effects from bubbles that may have been generated in bad minor loops.

Table 3-1. Power Down Sequence*

1. 13.5Vdc (X,Y coil down)
2. 17V (X,Y control logic)
3. +12V (Memory control current source)
4. +5V (Memory logic)
5. +12V (Control logic)
6. -12V (Control logic)
7. -5V (Control logic)
8. +5V (Control logic)

*Reverse the sequence for power up.

For reference, the power requirements of the prototype memory subsystem are approximately 27 watts in the quiescent state. During the 13ms dynamic period for a full memory (file) cycle the power peaks to approximately 80 watts. Table 3-2 summarizes the power supply requirement for the present prototype system. It is possible to decrease the power of the prototype system somewhat by using lower power equivalent devices in a few limited cases. Lower power consumption is expected to be possible in future operational system designs through the use of IC versions of the interface functions, and use of low power devices in the controller.

After the system and all parameters of the bubble control had been examined and determined to be operating within the latest requirements as specified by T.I., five memories were found to have various and easily detectable problems as follows:

a. Three were found to have bad minor loops, in addition to those found by T.I., yielding in excess of 13 bad loops.
b. One determined to be pattern sensitive.
c. One random errors $3/5 \times 10^{-5}$ errors/bit.

In addition, another unit was determined to have 1 more bad loop than was indicated by the map supplied with it. This would suggest that the T.I. testing process was failing to identify all of the bad loops in 25 - 30% of the units.

Also, two of the five units above exhibited decreased errors when the X and Y coil currents were increased above the maximum tolerance. Since the improvement in error rate was less than a factor of 10 it is difficult to conclude from this that there was a definite marginal operating condition. But, it does illustrate the sensitivity and complex relationships of the various factors involved.

Table 3-2. Power supply requirements

Voltage	Peak Current Drain (Amps)	Supply Capacity* Amps
+5	4.2	5
-5	.3	.5
-12**	.04	.1
+12	.2	.5
+17	.5	.75
+13.5***	4.0	5

*Design Spec. Recommendation

**Used only by 1702 PROM Map

***May become 12 volts in final T.I. versions.

In discussing these problems with T.I. they stated that they had recently found an inability of their test procedure to detect a specific type of problem. That is, devices in recent shipments were found to have inherent problems when stopping the bubbles in the minor loop because some bubbles tended to migrate. Our units may have had similar problems. They also emphasized how critical three of the control currents were. In order of priority they were identified as:

A. Transfer Out
B. Generate
C. Replicate

After reevaluating these characteristics, and satisfving T.I. that their specifications were being adhered to, they suggested that all units be returned for reevaluation. It was agreed that the above five units would be returned immediately, and the remainder might be returned pending the results of the final error rate tests. (The five units were returned with a detailed report shown in Appendix XII.) At this time it was decided to specify the final error rate tests based on knowledge gained from previous problems.

3.3 ERROR RATE TEST PROCEDURE

A set of test patterns was developed for conducting extensive error rate testing. The basic concept of the pattern designs is not only to test for pattern sensitivity, which was previously exhibited in a unit, but to also allow some discrimination as to what caused the errors, and identify where (major or minor loops) the sensitivity occured. A short description of the patterns is contained in Section 2.3. A more detailed description follows.

Eight patterns are generated consisting of 4 base patterns and their compliments. It is necessary to generate the compliment of a pattern to test each bubble position in each state for a given pattern. Each

pattern consists of 1, 2, or 4 words repetitively written into the 8 wide memory. Table 3-3 shows the patterns, and the test effect.

Two extensive tests were conducted. In each case the memories were read six times/write cycle. Error printouts were limited to 5/file/write cycle. The first test consisted of 2.7×10^9 bits read/system. A second test was conducted after the highest error rate memory boards were swapped with the lowest error rate boards. The objective here was to ascertain that the errors were a function of the memory board combinations, and not a unique characteristic of the system. This test consisted of 3.3×10^9 bits read/system.

A report was prepared for T.I. in order to aid them in their reevaluation of the units when returned at the completion of the error rate testing. This report is contained in Appendix XII.

3.4 DATA ANALYSIS

A summary of the computer error printout is contained in Tables 3-4 and 3-5, which were manually prepared in order to characterize the errors since the computer only prints an instant analysis on each error and not cumulative error analysis. Three types of errors were characterized:

1. Soft Errors - Error detected on a read pass, but not detected on a successive read pass.
2. Hard Errors - The same error detected on two or more successive read passes.
3. Migrating Error - This is a unique error type where an error was detected within the first 8 words, and would migrate towards the first word of a file on successive read passes. The error would then disappear after appearing in word Ø or 1. The exact migration characteristic appeared to be pattern related.

Table 3-3. Test patterns

Pattern	Word Ø	Word 1	Word 2	Word 3	Test Effect
1	00000000	11111111	00000000	11111111	Tests effect of alternating bubbles in major loops positions and minor loops.
2		Compliment of Pattern 1			
3	10101010	10101010	10101010	10101010	Test entire bubble in 1 state with high system noise.
4		Compliment of Pattern 3			
5	00000000	00000000	00000000	00000000	Same as pattern 3 & 4, but low noise.
6		Compliment of Pattern 5			
7*	ADD_H	ADD_L	$\overline{ADD}_H$	$\overline{ADD}_L$	Tests file integrity. Mixed major loop position and minor loop conditions.
8*		Compliment of Pattern 7			

*Based on file address (0-640)

ADD_H = High Byte

ADD_L = Low Byte

By determining the frequency of errors in each pattern it could be shown that a definite pattern sensitivity characteristic existed. Approximately 55% of the errors detected occurred during patterns 1 and 2. This indicated a sensitivity to patterns where alternate minor loops had a large majority of bubbles interleaved with minor loops having a large majority of absence of bubbles. Another 30% of the errors occurred in patterns 7 and 8. Since these patterns are file address based, it was necessary to determine the minor loop status in the area of the errors. It was found that patterns similar to those produced by patterns 1 and 2 existed in the minor loops around the loop in which the error was found. The remaining 15% of the errors occurred in patterns 3 - 6. Two other characteristics were immediately obvious during the data analysis:

A. 90% of the hard errors occurred on the first read pass whi would indicate a marginal write operation.
B. 85% of the errors were the result of gaining a bubble.

Comparing the results from the two tests separately it can be seen that in 3 of 4 cases where the boards were swapped that the high error rates followed the boards. In the fourth case the high error rate stayed with the bit position in the system. However, the errors in the latter case have the same general characteristics as the others. This tends to decrease the possibility that a unique position related problem exists in the system, and tends to suggest that there may have been some slight differences in the tests. These differences might include ambient temperature, slight voltage change, or negative effects from collapsing the memory fields.

Table 3-6 is a composite of Tables 3-4 and 3-5. The overall results of the two tests are summarized on the following page.

System Error Rate	5×10^{-8}	errors/bit
Unit Error Rate - High	2×10^{-8}	errors/bit
- Low	1.6×10^{-10}	errors/bit
Soft Error Rate	3×10^{-9}	errors/bit
Soft Migrating Error Rate	4.7×10^{-9}	errors/bit
Hard Error Rate	4.3×10^{-8}	errors/bit
Percent of Hard Errors detected on 1st read	87%	
Percent of Error due to bubble gained	85%	

It is important to note that although these rates are higher than the specified soft error rate of 10^{-9} errors/bit and hard error rate of 10^{-10} to 10^{-11} errors/bit, they would be acceptable in data acquisition system applications. There did not appear to be any failure modes (other than those that could be controlled such as the power on sequence) that would cause random data scrambling.

3.5 TEXAS INSTRUMENT REEVALUATION OF BUBBLE MEMORIES

On 19 September 1977 a verbal report was received from Gerald Cox of the Magnetic Bubble Memory Product Marketing group for T.I. The report consisted of only general statements and not specific error rates, error characterization or errors relative to specific units by serial number. Their reevaluation test consisted of individually testing 11 of the 14 units returned, for approximately 5 minutes, by alternately writing and reading a "simple shift pattern." Their results and comments on the results were as follows:

1. One unit had a low threshold sense amplifier (Motorola device) on the memory board. They had previously identified this problem, and are presently producing their own chip with a minimum guaranteed sense threshold.

2. One unit exhibited no errors.
3. The remaining 7 memories, mounted on boards, and 2 of the original 5 units returned exhibited problems due to the generate process. The characteristic of the error was that extra bubbles were being generated, an effect they had only recently discovered. This problem can be reduced or eliminated by reducing the generate current pulse in both width and amplitude since only 50 nanoseconds are required to reliably generate a bubble. (Note that the present T.I. timing design only allows 250 micro-second resolution, and this would require some design changes.) They stated, however, that the real solutions to eliminate the problem were 1) more stringent material control, and 2) a redesign of the generate method.
4. No explanation was given for not testing the remaining 3 units.

Only a general correlation between Develco's test results and T.I.'s evaluation and previous inputs is possible since T.I. did not maintain records as to the error rate relative to unit serial number. However, their findings confirm our test results in several areas.

A. They found 1 unit without problems which confirms our finding.
B. They found 1 unit in the last 9 returned with a sense amplifier problem. This agrees with our results in that the errors on 1 unit not used in the final tests could not be easily characterized.
C. Seven of the units from the final tests exhibited generate problems which created excess bubbles. This generally correlates with the test data in that 85% of all errors were hard, and 87% of those errors resulted in a gained bubble, and occurred on the 1st read pass.
D. They found 2 units with suspected additional bad loops. Our tests indicated a total of 4 units. However, they did not test 3 of the first 5 units returned of which we had found 3 that had additional bad loops.

E. During the evaluation, they suggested the possibility that our units might have problems related to stopping the bubble cycle. This may explain the remainder of the hard errors which occurred after the 1st read pass.

F. No explanation or correlation has been determined for the migrating errors.

G. Possible explanations of the soft error could be many in number, but in most cases would be confined to noise and/or signal to noise margin of the sensing circuitry. Another possibility is that some errors may result if nominal timing values are not optimum.

After completing their tests, T.I. agreed that there are performance problems in the prototype devices, both those they are currently making and those purchased for evaluation on this program, which are due to design and manufacturing problems. As a result of their evaluation of the devices we used, T.I. recalled all the MBM's used in this program. It is clear that T.I. has some more development work to do before their MBM's reach production status. However, it is extremely important to recognize that:

a) This is a new and radically different technology, so a longer learning curve can be expected.

b) T.I. took a calculated risk, but we believe a worthwhile one, in releasing prototype versions at this stage of development so the industry could learn to use them.

c) The error rates that resulted from the prototype units used in the final tests would have been entirely acceptable in many data acquisition system applications.

d) T.I. will undoubtedly solve the device problems, and it is our opinion that refined devices will be back on the market by the 1st quarter of 1978.

Table 3-4. Test results 5 August 1977

2.7 x 10^9 bits/system (3.38 x 10^8 bits/board.)

TBM S/N	BRD S/N	Bit Position	Failures vs. Pattern								Total Failures	Types of Failures			Hard Errors on 1st Read	Bubble	
			1	2	3	4	5	6	7	8		Soft	Hard	Error Shift		Lost	Gained
72-53-11	09	0	5	1	1	0	2	0	1	3	13	2	10	1	6	4	9
74-36-08	05	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74-36-01	01	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78-13-15	08	3	9	9	0	0	0	0	5	11	34	0	33	1	33	1	33
54-24-02	02	4	7	13	1	0	0	0	7	15	43	0	40	3	38	3	40
72-63-06	07	5	4	4	5	0	1	2	0	4	20	0	19	1	12	2	18
72-63-04	06	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
68-94-00	03	7	0	1	0	0	1	0	0	0	2	0	1	1	1	1	1
		TOTAL	28	28	7	0	4	2	13	33	112	2	103	7	90	10	101

91%

Table 3-5. Test Results 15 August 1977

3.32×10^9 bits/system (4.15×10^8 bits/board)

TBN S/N	BRD S/N	Bit Position	Failures vs. Pattern								Total Failures	Types of Failures			Hard Error on 1st Board	Bubble	
			1	2	3	4	5	6	7	8		Soft	Hard	Error Shift		Lost	Gained
74-36-01	01	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78-13-15	08	1	20	12	0	1	0	0	5	10	48	1	44	2	44	4	44
72-53-11	09	2	1	4	1	0	2	0	0	0	8	2	5	1	1	3	5
74-36-08	05	3	0	0	1	1	0	2	0	0	4	4	0	0	0	1	3
72-63-04	06	4	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1
68-94-00	03	5	23	18	0	5	6	0	0	0	52	4	40	8	24	12	40
54-24-02	02	6	16	20	6	0	7	0	2	12	83	5	67	10	66	14	69
72-63-06	07	7	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1
		TOTALS	60	54	8	7	15	2	7	22	197	17	158	21	137	34	163

Table 3-6. Test results composite of tables 3-4 and 3-5

6.02×10^9 bits/system (7.53×10^8 bits/board)

TBM S/N	BRD S/N	Bit Position		Failures vs. Pattern								Total Failures	Types of Failures			Hard Errors on 1st read	Bubble	
		1st Test	2nd Test	1	2	3	4	5	6	7	8		Soft	Hard	Error Shift		Lost	Gained
74-36-01	01	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78-13-15	08	3	1	29	21	0	1	0	0	10	21	82	1	78	3	77	5	77
72-53-11	09	0	2	6	5	2	0	4	0	1	3	21	4	15	0	0	0	0
74-36-08	05	1	3	0	0	1	1	0	2	0	0	4	4	0	0	0	1	3
72-63-04	06	5	4	1	0	0	0	0	0	0	0	1	0	1	0	1	0	1
68-94-00	03	7	5	23	19	0	5	7	0	0	0	54	5	40	9	25	13	41
54-24-02	02	4	6	23	33	7	0	7	0	29	27	126	5	108	13	104	17	109
72-63-06	07	5	7	5	4	5	0	1	2	0	4	21	0	21	1	13	2	19
		TOTALS		88	82	15	7	19	4	40	55	309	19	261	28	227	44	264

SECTION 4 - DATA ACQUISITION SYSTEM DESIGN RECOMMENDATIONS

It is clear from our investigations that MBM technology is progressing rapidly, and that even today's prototype versions can be successfully used in field data acquisition applications. The relatively crude architecture and lack of a full family of peripheral components for presently available MBM devices makes design difficult and limits flexibility. However, advantages such as nonvolatility, ruggedness, and relatively low power consumption make these devices useful even in todays technology. Thus, we recommend that some first generation system designs be developed now to take advantage of MBM features in special applications, and gain operational experience. It then appears that the technology will reach a level of maturity in time to coincide with the development of second generation systems for broad based general use in the next 1 to 2 years.

As an example of the progress being made, T.I. has already incorporated MBM material and generate pulse specification changes to eliminate the major error source problem we encountered as described in Section 3. Although the changes are probably not final, prototype units can now be obtained (with a 16 to 20 week delivery) that have satisfactorily improved performance. At present T.I. expects to be in production within the year, however, details on price and delivery are not yet available.

There is also considerable progress in the area of support chips. T.I. has had an I.C. version of a controller available for some time, but it does not appear to be generally useful in data acquisition applications particularly for 8 bit wide storage. It is conceivable the controller could be of use in limited applications whether for single channel serial recording or where a poling arrangement can be tolerated in parallel applications. Although some consideration has been given to an 8 wide, or byte, memory controller, it is not expected one will be available in the near future.

However, the development of other support devices is nearing completion, and it is expected that they will be available by late in the first quarter of 1978. The custom IC's that are now undergoing family tests include seven IC's for use with each MBM and one IC for each system as follows:

1. One 16 pin DIP (Dual-Inline Package) with the 5 control current sources including temperature compensation (range not specified) and the Δ current/Δtemperature requirements for the memory.

2. Two 8 pin DIP's with the coil drivers. and two with the diode packs for the coil drivers.

3. One 8 pin DIP with a sense amplifier that has guaranteed threshold detection and the required output logic. This will also be available in a 14 pin DIP that provides a tri-state output. An additional 8 pin Hybrid containing the sense amplifier RC, etc., components is also required.

4. A single 22 pin DIP is being developed to replace all of the function timing circuitry with the exception of an external oscillator. This will have a fanout capacity of 17 T^2L loads which will allow it to be used directly in an 8 wide system.

The system design requirements for 3 cases of first generation data acquisition systems using MBM's have been considered as discussed in the following sections. The first is based on utilizing the prototype memory model which was designed to explore the advantages of doing the control operations in real time. Although the expected benefits did not materialize, because of the complexities of the major-minor loop operating requirements and the fact that storage in the major loop is not possible in present T.I. devices, it can certainly be used in a data system with a minimum of additional development. An alternate configuration, also based on the T.I. MBM, was considered to achieve a simplified controller design, more

flexibility for strobed operation at lower data rates, and lower standby power. Finally, a straight serial recorder was considered that would take advantage of the simplicity possible with the single loop memory such as that available form Plessey.

4.1 DATA ACQUISITION SYSTEM USING 8 BIT PARALLEL PROTOTYPE MEMORY

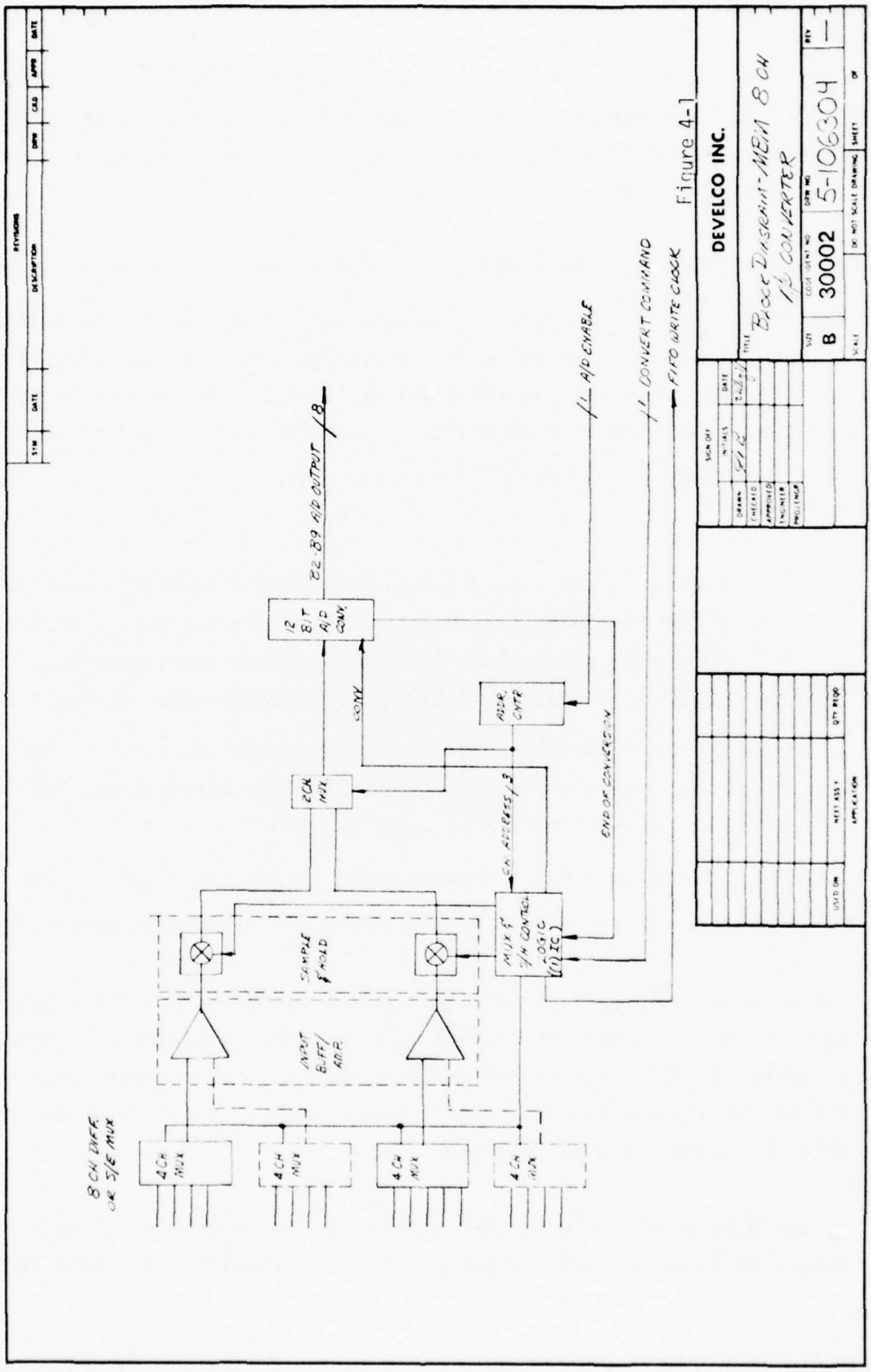
The Prototype Magnetic Bubble Memory system was designed (in terms of both hardware and software) to be easily used in a data acquisition system. However, several additions will be required to bring the system from its present status to a field usable piece of hardware. This would include the following steps:

1. Add an A/D converter board which could be the same as a modification of the Mighty Epic Underground Telemetry System 2 channel multiplex input and boards. Of course, other configurations, such as multiple channel multiplexing, are possible, Figure 4.1.
2. Prepare the additional software required which consists of bubble controller programs for multiplexer and A/D operation and PDP-11 Fortran programs for data printout operation.
3. Add the necessary battery power supplies for field use.
4. Add a D/A converter for field checkout and quick look capability.

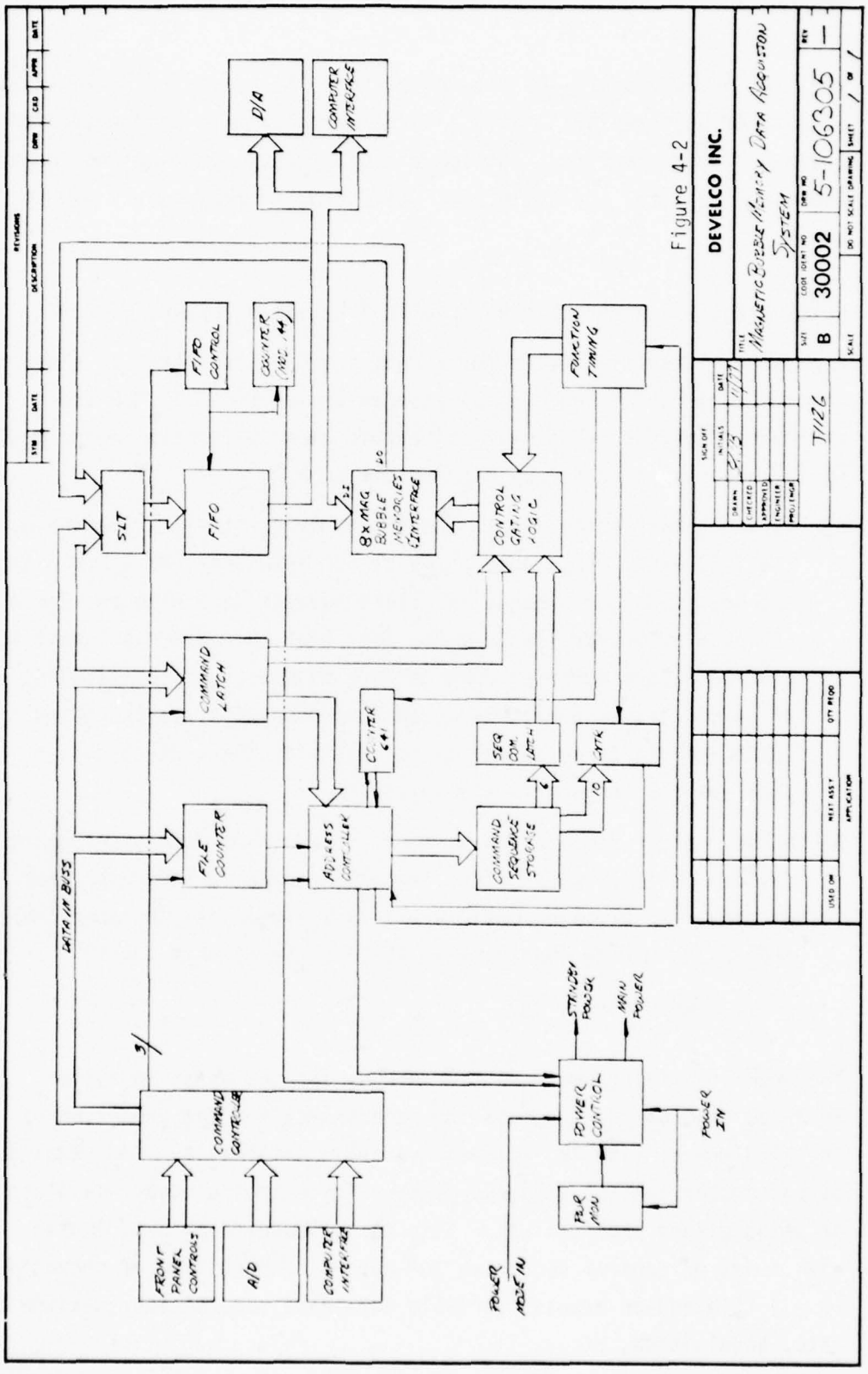
These steps would yield a data acquisition system with a fixed data rate of 50k bytes/sec. with a 92k byte capacity, allowing a 12.5kHz response (@-3dB). The system would require a constant power source, though provision will be made for power up/down control prior to and after all data has been recorded.

It would be possible to further modify the system to also allow variable data rate input of $\leq$ 20k bytes/second. To accomplish this each FIFO

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC



THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC



register would require 256 bits of storage (64 presently) allowing the data to be buffered until a full files worth of data were ready for storage. Some additional control circuitry would be required, however, 256 bit FIFO's are available that will directly replace the 64 bit units.

4.2 ALTERNATE 8 BIT PARALLEL DIGITAL DATA ACQUISITION DESIGN CONCEPT

A possible alternate design for a data acquisition system is shown in block diagram 4-2. The concept assumes use of the T.I. MBM and peripheral devices. The main objectives of an alternate design would be to produce a data acquisition system with

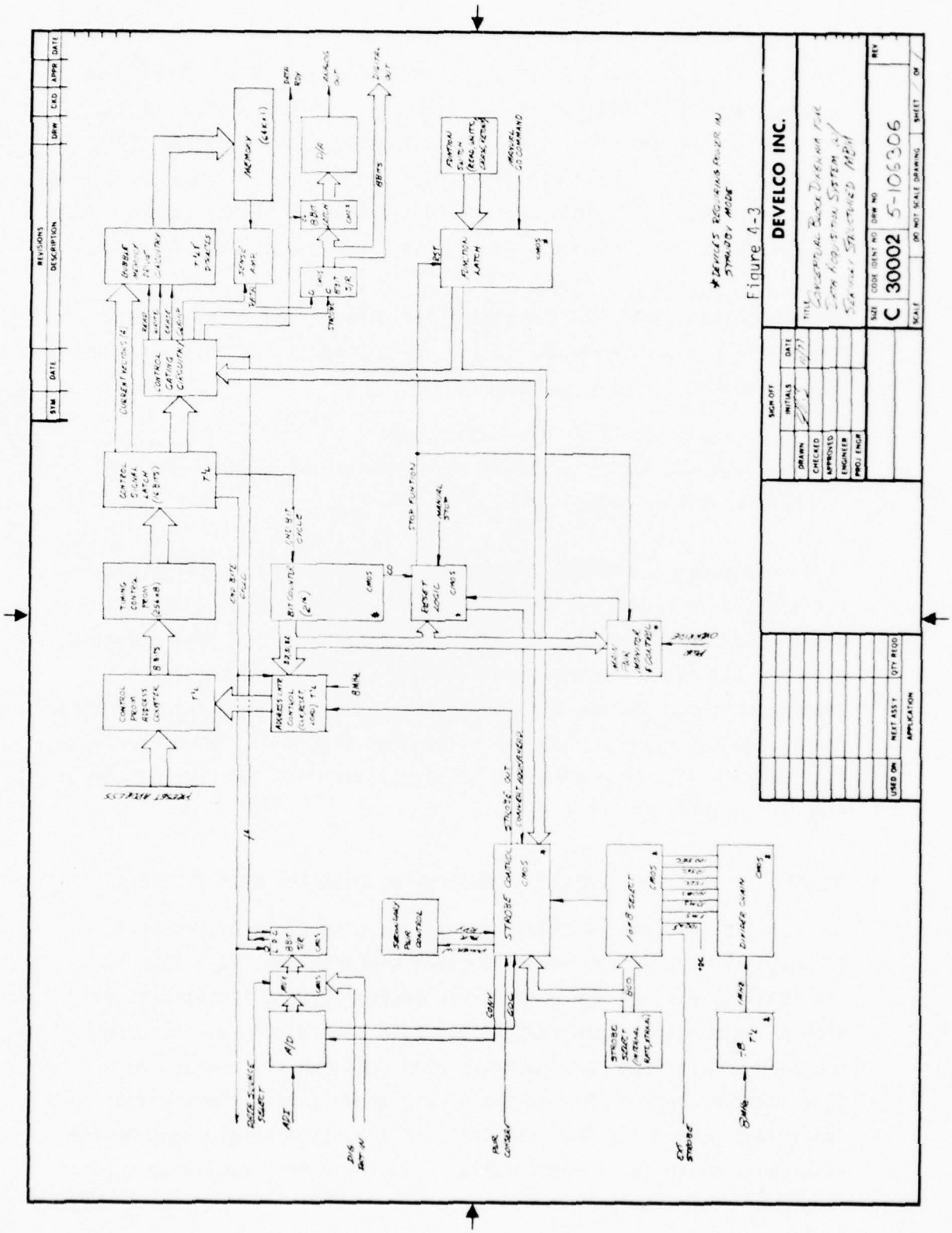
A) Variable data rates within the limits or constraints imposed by the T.I. unit, e.g.; data rates of 50k bytes/sec. or $\leq$ 20k bytes/second. (The gap between the fixed maximum data rate and lower variable rates results from the fact that the major loop must be cleared before powering down between samples. The possibility of permitting such storage under some conditions is being investigated, and if it turns out to be possible, the control design problem will be greatly simplified.)

B) Minimum power consumption by means of powering down unnecessary devices during periods of not recording data at low data rates. By properly selecting devices, it is believed standby power can be reduced to <100mW (possibly at the expense of chip count).

C) Simplified control logic.

The function timing would be the same as for the existing system. The FIFO control would remain the same expect for the expansion of the FIFO memory size to implement variable data rates. The remainder of the system concept changes. The basic control concept would consist of using direct PROM control. This is implemented by programming PROMS with a set of control sequences for each different type of operation (e.g.; file add-on access, variable data rate acquisition, maximum data rate, acquisition, etc.).

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC



The base address of the PROM is a result of the function input. A command consisting of Annihilate, Generate Replicate, (etc.) bits, and count value would be clocked from the PROMS to registers. The counter would be decremented every 10μs bubble cycle. When the counter was decremented to 0 the next command and count would be loaded. This process would continue until an entire sequence had been executed.

The controller could be commanded by a computer for test and data retrieval or commanded manually for field operation. A file number counter would serve two purposes;

1. To access a specific file number, and
2. To count the number of files recorded in field operations for automatic shut-down.

In a data acquisition mode the entire system would be powered when 144 bytes were stored in the FIFO. The next sequential file in the minor loops would be transferred out, annihilated, and then recorded with the 144 bytes. Other data could be strobed into the FIFOS simultaneously. The new file would then be transferred into the minor loops, bubbles stopped, and the system would go into standby power mode. This process would repeat unitl all 641 files were recorded at which time all power would shut down.

4.3 CONTROL CONCEPT FOR A SERIAL ARCHITECTURE MAGNETIC BUBBLE MEMORY

The parallel digital recorders described in Sections 4.1 and 4.2, although they have important potential applications, lack the flexibility required for general use because of the limitations of present major-minor loop devices for data recording use. Although the major-minor loop configuration will continue to receive the most emphasis because it has the widest application, there may be sufficient demand for the simplicity of Plessey's single loop device to keep it alive for several years until the others evolve to a point

where they are easier to use. (We have just learned that Rockwell has dropped its serial configuration, at least in commercial versions. So far, Plessey is advertizing plans to offer both kinds, but more inquiries are necessary to confirm their long range goals.) We have, therefore, considered a simple recorder type design, discussed below, that sacrafices speed for flexibility and lower cost in an initial attempt to achieve a general purpose machine.

The concept for a single serially structured bubble memory in a data storage system is illustrated in the block diagram, Figure 4-3. The concept assumes that data will be stored in serial units of 8 bit words (bytes), and as a result of this the integrity of a byte boundry must be maintained. The concept provides four main functions:

1. Read
2. Write
3. Erase
4. Return (to start point)

An exclusive function is latched into memory by a go command which executes the command. A command will be terminated at the end of the memory or when an external stop is commanded (recognized on byte boundaries).

In the read or write mode the data (byte) rate is variable. A byte cycle may be stimulated from 1 of 7 internal selectable rates ranging in decade steps from 1 byte/100μs to 1 byte/100 sec. A cycle or groups of cycles may also be stimulated from an external strobe source. The fastest byte cycle rate is obtained by maintaining a logic 1 on the strobe control circuitry.

If it were desirable to store blocks of data consisting of more than 1 byte, the END BYTE signal could be used to control a multiplexer. After

the number of bytes in a block had been counted the external strobe would be returned to logic "0" until the next sample was desired.

If an internal byte rate were selected, all noncritical devices would be powered down at rates $\leq$ 1 byte/100ms. This obviously would reduce standby power and allow extended field operations on a battery power source.

When in the external strobe mode an external input would be required to cause the system to go into standby power mode when the strobe cycle had been completed.

The control currents for the memory are provided by the memory drive circuitry which would provide seven current signals to the memory. The timing of the signals would be programmed into a 256 x 8 bit PROM. Sixteen timing channels would be available by interleaving 8 channels in alternate PROM addresses. Some of the remaining 9 timing channels would be used for control timing, and some of these would also be gated depending on the function.

An 8 bit counter, operating at 8mHz, would provide the addressing control for the PROM yielding 1/4μs resolution in the timing controls. The PROM outputs would be stored in two 8 bit latches, each receiving data from alternate PROM addresses. Some of the control signals would be gated to the memory drive circuitry dependent on the function signals.

A 16 bit counter would be used to count the number of bits processed allowing end of memory detection. The 3 least significant bits would be used to cause timing cycles in groups of 8 bits, thus maintaining byte boundaries.

The system described in the diagram also provides for both digital and analog I/O interfaces. This would provide for either mode of operation

or both where it might be desirable to interleave both types of data. The main power control would shut down the entire unit after the entire memory had been accessed if the power override were turned off after the system was initialized. The system would also automatically shut down if the power source dropped below a level where improper operation might effect data in the memory.

The block diagram may be simplified as follows:

A. Limit the data rate to an external source eliminating approximately 10 IC's.

B. Eliminating power control circuitry.

Further simplification in the control circuitry could be realized by producing an 8 bit wide system. This would require the memory drive circuitry to be produced 8 times which offsets the savings in control complexity, but the maximum throughput would increase by a factor of 10.

REFERENCES

1. Magnetic Bubble Technology: Integrated Circuit Magnetics for Digital Storage and Processing, Edited by Hsu Chang, IEEE Press, 1975.

2. Foreign and Domestic Accomplishments In Magnetic Bubble Device Technology (1967 - 1975), Robert B.J. Warnar and Peter J. Calomeris, NBS Special Publication, May 1977.

3. Magnetic Bubble Memories, Anon., PP50-66, Digital Design, May 1977.

4. Magnetic Bubble Systems Approach Practical Use, J. Egil Juliussen, Computer Design, October 1976.

5. Designing a Magnetic Bubble Data Recorder, Part 1 & 2, Eric J. Hoffman, Robert C. Moore, and Terrence L. McGovern, Computer Design, March and April, 1976.

6. Limitations of and Expectations for Magnetic Bubble Memories, R.B. Clover, Hewlett-Packard Co., Palo Alto, CA, Spring Compcon 77 Digest of Papers.

7. PDP-11 Peripherals Handbook (Programming, Interfacing), Digital Equipment Corporation.

8. PDP-11/05,10,35,40 Processor Handbook, Digital Equipment Corporation, 1973.

APPENDIX I

TEXAS INSTRUMENTS EQUIPMENT SPECIFICATION

TBM0101 BUBBLE MEMORY DEVICE

A

APPLICATION		REVISIONS			
NEXT ASSY	USED ON	LTR	DESCRIPTION	DATE	APPROVED
		B	Overall control timing explanation	10/27/75	RAN
		C	Wide temp range material	9/1/76	RAN
		D	Change detector polarity	11/15/76	G M C
		E	Low current coil implementation	02/04/77	D J H

TENTATIVE DATA SHEET

THIS DOCUMENT PROVIDES TENTATIVE INFORMATION ON A NEW PRODUCT. TEXAS INSTRUMENTS RESERVES THE RIGHT TO CHANGE SPECIFICATIONS FOR THIS PRODUCT IN ANY MANNER WITHOUT NOTICE.

REV STATUS OF SHEETS																		
REV	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	
SHEET	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES
ANGLES ±1°
2 PLACE DECIMAL ± .010
3 PLACE DECIMAL ± .002
IDENTIFYING NUMBERS SHOWN IN PARENTHESES FOR REFERENCE ONLY
INTERPRET DWG IN ACCORDANCE WITH MIL STD 100

DWN G. Sloat 10/17/75
CHK
ENGR
QA
APVD R. Naden 10/17/75
CONTR NO
DESIGN ACTIVITY RELEASE

TEXAS INSTRUMENTS
INCORPORATED
Equipment Group Dallas, Texas

EQUIPMENT SPECIFICATION
TBM 0101 BUBBLE MEMORY DEVICE

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651

SCALE | SHEET

1.0 SCOPE

This specification describes the electrical, mechanical and environmental characteristics of the TBM 0101 Bubble Memory Device.

2.0 GENERAL DESCRIPTION

The TBM 0101 Bubble Memory Device is a 14 pin dual in-line module containing a 92 K bit bubble memory chip, coils for providing a rotating magnetic field, a permanent magnet structure for providing the required static magnetic field, and a magnetic shield assembly.

3.0 FEATURES

92K bit nonvolatile memory
Occupies less than 0.5 cubic inch
Consumes less than 0.7 watts for continuous operation
Weighs less than 25 grams
Rugged package with self-contained magnetics
Major/minor loop architecture
Average access time (first bit) - 4 ms
100 KHz read/write operation
50 Kb/s data rate

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV A	SHEET 1 of 17

TI-7915C

4.0 APPLICABLE DOCUMENTS

92 K chip specification TI Drawing No. SK 2007653

TMS 5502 Bubble Memory Controller Specification TI Drawing No. SK 2007654

Schematic - Bubble Memory Card TI Drawing No. SK 2007655

5.0 FUNCTIONAL DESCRIPTION

5.1 Bubble Chip Operation

The 92 K bit bubble chip consists of a substrate of gadolinium gallium garnet on which is grown an epitaxial film of magnetic garnet material. This film supports 5 μm cylindrical bubble domains.

Patterns of permalloy metal are deposited on the epitaxial film to define the path of the cylindrical bubble domain in the presence of a rotating magnetic field. As the field rotates, the bubble domains move under the metallic patterns in shift register fashion.

A static magnetic field (bias field) is provided perpendicular to the plane of the chip to control the size and shape of the bubble domains.

Control functions (i.e. generate, transfer in, transfer out, replicate and annihilate) are accomplished by providing current pulses through the appropriate control elements on the chip. Current flowing through a control element causes a local alteration in the magnetic field. This field alteration, in conjunction with the local permalloy pattern, accomplishes the control function.

After generation, the bubble domains move 22.5 μm during each rotation of the magnetic field. Control functions must be timed to ensure that the bubble domain has arrived at the proper location on the chip.

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV A	SHEET 2 of 17

TI-7913C

To detect the presence or absence of bubbles, two magneto-resistive elements are provided. These elements may be used as part of a bridge circuit configuration which provides a high degree of noise cancellation when used with an external differential amplifier.

5.2 Major/Minor Loop Operation

See Figure 1.

Bubble domains representing bits may be created under the generate loop. These domains step through the major loop in the indicated direction.

When a data string has been generated, equal in length to the number of minor loops (a page), and shifted such that the first bit is positioned over the first minor loop, the transfer gate may be energized. This operation transfers the page from the major loop into the minor loops. The data now circulates in the minor loops. New data may be generated, shifted, and transferred into each of 641 different minor loop page positions.

Data is retrieved by doing a transfer out operation when the desired page rotates to the top of the minor loops. This moves the page out of the minor loop structure and puts it back into the major loop. The page now moves around the major loop in bit serial form until the first bit arrives at the replicate/annihilate element. One of two control operations may now be performed:

5.2.1 Replicate

If a replicate pulse is issued each time a bubble domain arrives at the replicate/annihilate gate, each domain is stretched and cut. One portion is diverted into the detector area to be read and the other continues

SIZE	CODE IDENT NO	DRAWING NO
A.	96214	SK 2007651
SCALE	REV A	SHEET 3 of 17

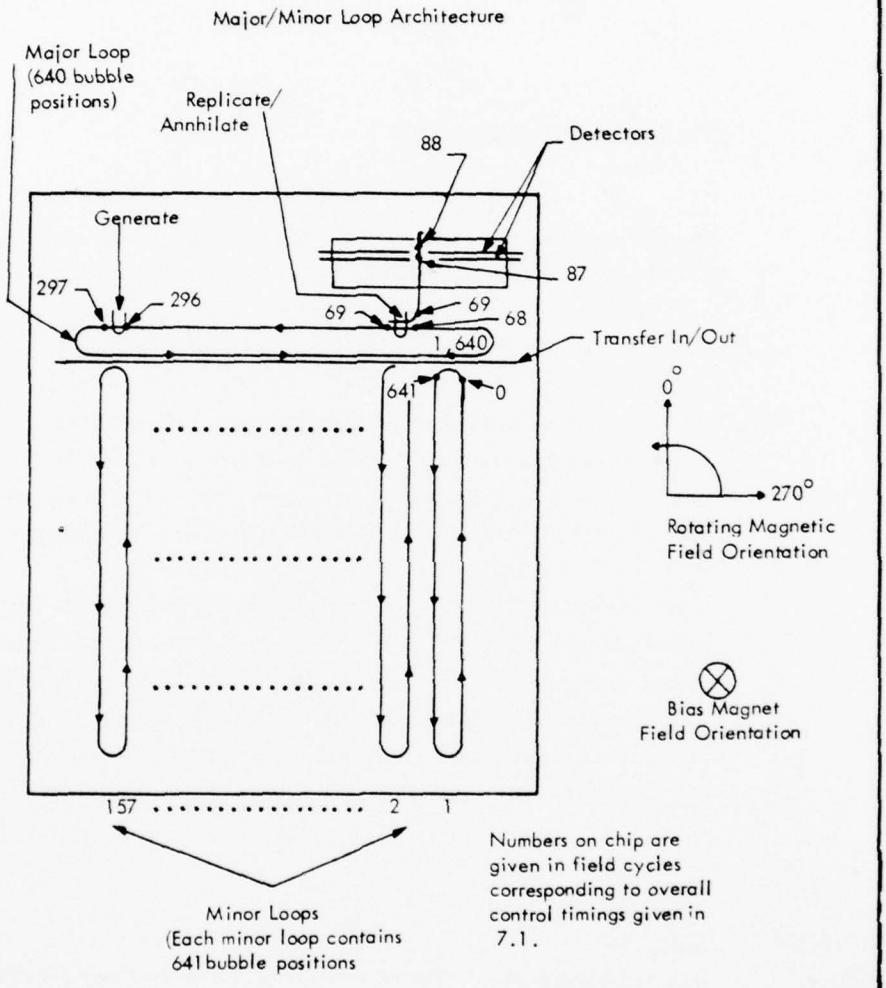


Figure 1

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV	SHEET 4 of 17

moving in the major loop. It is thus possible to read a page, recirculate the information around the major loop and put it back into the minor loops for non-volatile storage. This operation provides a nondestructive read capability.

5.2.2 Annihilate

Annihilation is accomplished by applying a current pulse to the replicate/annihilate gate that transfers the bubble out of the major loop and into the detector track where it propogates off the chip.

5.3 Redundancy

To enhance production yields and reduce device cost, 144 of the 157 minor loops are guaranteed to be useful. The user's system must insure that information is not written into defective loops and it must ignore any information read from these loops.

Defective minor loops result from manufacturing processes related to the small geometry of the permalloy patterns. Defective minor loops are identified during factory testing and in no way relate to a field failure mode. A map of the defective minor loop location will be written into the device at the factory before shipment. This map must be read by the user in order to implement data transfers to the device.

5.4 Device Capacity

Minor loop count	(total)	157
Minor loop count	(useful)	144
Minor loop count	(unused)	13

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV	SHEET 5 of 17

TI 7915C

5.4 Device Capacity (Continued)

Major loop length	640	periods
Minor loop length	641	periods
Raw data capacity	100,637	bits
Useful data capacity (92 K bits nominal)	92,304	bits
Percent redundant storage	8.28	percent

Redundancy shall be implemented by avoiding the writing of bubbles into any defective loop and neglecting bubble information read from any defective loop.

5.5 Performance Specifications at 100 KHz Operation

	Min	Nom	Max	Unit
Average access time (first bit)	-	4	-	ms
Average cycle time (144 bit page)	-	9.6	-	ms
Hard error rate after N years residence time with operating duty cycle d **	-		10^{-9} Nd	errors/bit
Soft error rate*			10^{-9}	
Data rate	-	50	-	Kbits/sec

*Depends upon sense electronics and layout

**This maximum error rate shall be sustained on reading data that has been resident in the device for Nd years of continuous READ/WRITE operation, e.g., for N = 1 and d = 1/100, hard error rate = 10^{-11} errors/bit; for N = 0.1 and d = 1, hard error rate = 10^{-10} errors/bit.

5.6 Bias Margin

The chip shall have an operating bias margin of no less than 8 Oe. This minimum margin shall be achieved under any combination of the parameters

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV	SHEET 6 of 17

TI 7913C

5.6 Bias Margin (continued)

specified herein, i.e., temperature, drive field, control pulse current amplitude and timing, humidity and vibration.

6.0 ELECTRICAL

6.1 Coil Drive

The bubble chip is surrounded by two orthogonal coils which, when driven 90° out of phase, produce a rotating magnetic field in the plane of the bubble chip. It is this rotating field that causes bubble domains to move under the permalloy patterns.

If the two coils are turned on and off in the prescribed manner, bubble domain motion can be started and stopped without error. Thus, if continuous duty is not required, a power savings can be realized by operating in the start/stop mode.

Triangular current waveforms are recommended for driving the two coils. This approach lends itself to precise digital control of the drive field.

See Figure 2.

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV	SHEET 7 of 17

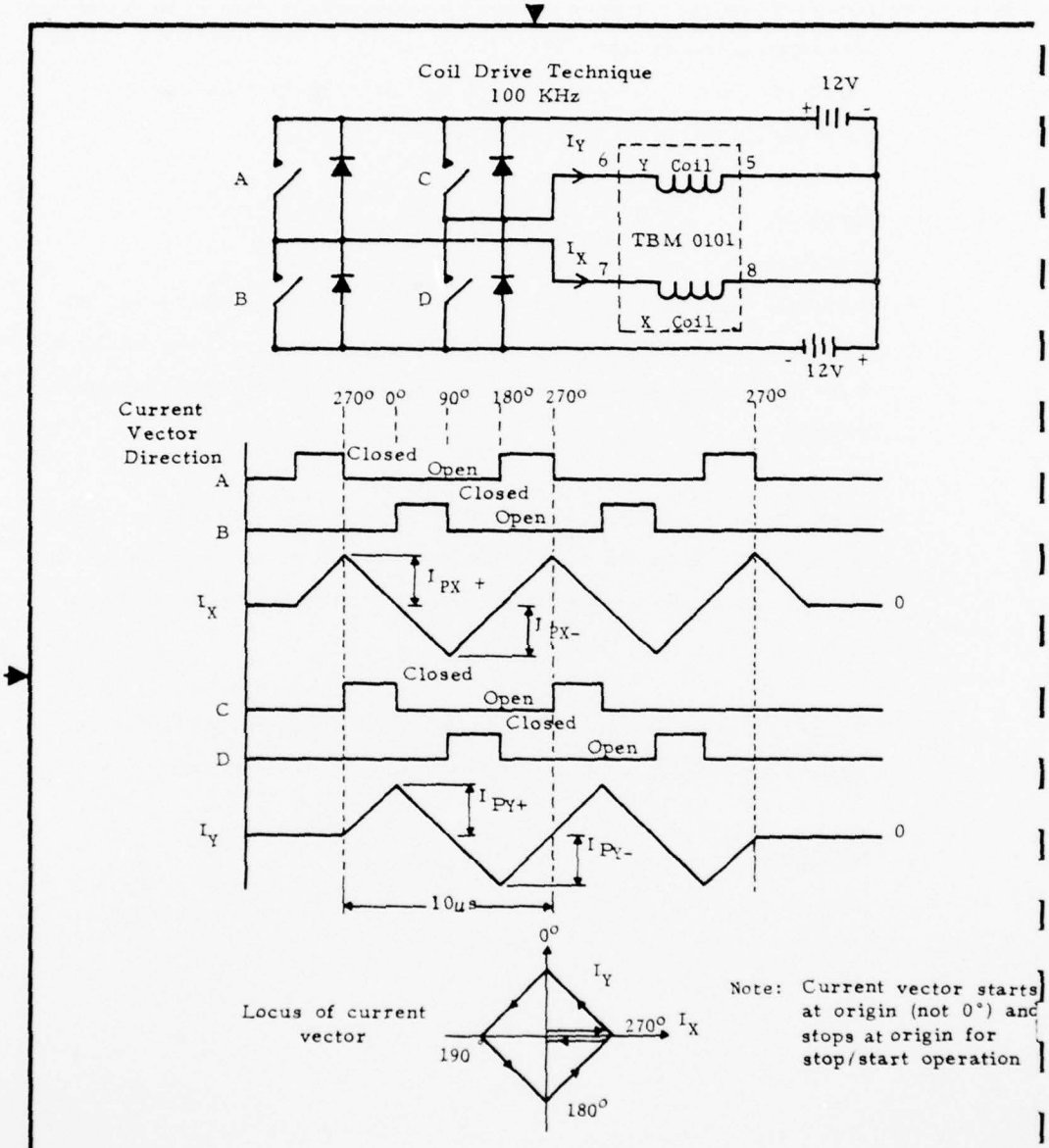


Figure 2

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV A	SHEET 8 of 17

6.1.1 Coil Drive Requirements

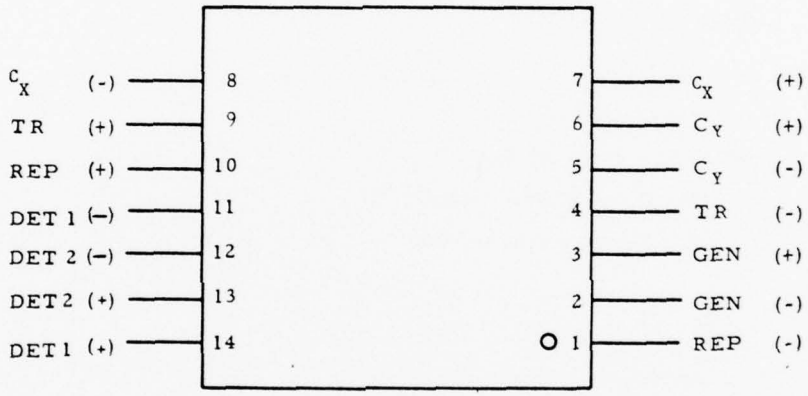
	Min	Nom	Max	Unit
Peak field strength for triangular drive	48	50	54	Oersteds
Field rotation frequency	99.99	100.00	100.01	KHz
Phase lag(inner coil with respect to outer coil)	87	90	93	Degrees
Total coil power		0.7	1.1	Watts
X Coil				
Inductance	62	64	66	µh
Resistance (DC)	-	3.9	-	Ohms
Resistance (AC at 100 KHz)	-	4.8	-	Ohms
I_{PX+} (Positive peak current)	445	460	485	ma
I_{PX-} (Negative peak current)	445	460	485	ma
Y Coil				
Inductance	60	62	64	µh
Resistance (DC)	-	3.0	-	Ohms
Resistance (AC at 100 KHz)	-	4.3	-	Ohms
I_{PY+} (Positive peak current)	465	480	495	ma
I_{PY-} (Negative peak current)	465	480	495	ma

7.0 CONTROL CURRENT TIMINGS AND AMPLITUDES

All timings are based on the direction of the rotating field with respect to the bubble device orientation. These timings are delays from the time when the field

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV	SHEET 9 of 17

Figure 3 Pin Assignment



SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV A	SHEET 10 of 17

TI-7915C

	Leading Edge (μs)			Pulse Width (μs)			Amplitude* (mA)		
	Min	Nom	Max	Min	Nom	Max	Min	Nom	Max
Generate	0.125	.25	.375	0.125	.25	.375	240-1.2T	250-1.2T	280-1.2T
Replicate	5.125	5.25	5.375	1.375	1.50	1.625	125-.57T	135-.57T	145-.57T
Annihilate	3.375	3.5	3.625	4.625	4.75	4.875	40	60	80
Transfer Out	0.875	1.0	1.125	1.625	1.75	1.875	40-.09T	42-.09T	44-.09T
Transfer In	4.625	4.75	4.875	2.375	2.5	2.625	25	27	30
Detector	DC	DC	DC	DC	DC	DC	4.8	5.5	6.2
Clamp †	.625	.75	.875	1.625	1.75	1.875		TTL	
Strobe** †	1.625	1.75	1.875	0.125	.25	.375		TTL	

*T is degrees Celsius

** Strobe is an edge sensitive function and is defined for a positive edge

† Suggested timings only; See 9.0.

SIZE A | CODE IDENT NO 96214 | DRAWING NO SK 2007651 | SCALE | REV | SHEET 11 of 17

7.0 CONTROL CURRENT TIMINGS AND AMPLITUDES (Continued)

is 0°. The current conventions are in accordance with Figure 3.

7.1 Control Timings (Overall)

The following timings refer to the first bubble to be read or written in a page of bubble data. A "page" is the block of information obtained from the transfer operation, i.e., one bit from each of the 157 loops. The timings are taken from the leading edge of the first control pulse to the leading edge of the second control pulse. These timings are used only to verify the overall control sequence and should not be used to verify the timings within each drive field cycle as given in 7.0 . The timings are based on use of a 100 KHz drive field accurate to 0.01% in frequency. If the user's drive field frequency F (KHz) differs from 100.00 KHz his timings should be multiplied by the factor F/100 before comparison with these values.

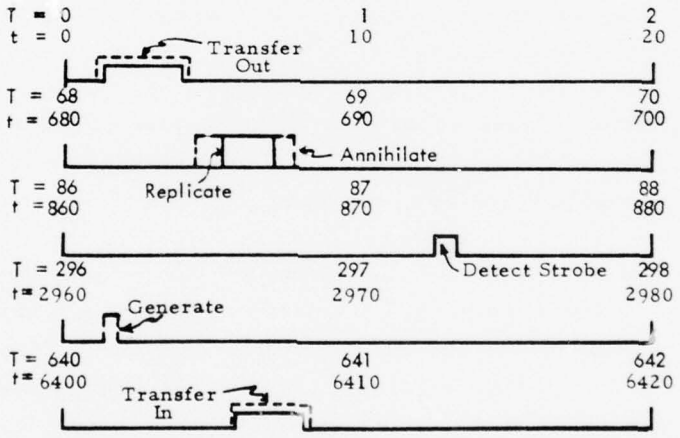
Transfer out to replicate	684	±1 μs
Transfer out to bubble signal peak	872	±1 μs
Transfer out to generate	2962	±1 μs
Transfer out to transfer in	6405	±1 μs

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV	SHEET 12 of 17

TI-7915C

7.1.1

The bubble functions are shown schematically on the following diagram to show the operations necessary to read (solid lines) or write (dashed lines) the first bit in a page (the bit from minor loop number 1). The minor loops are spaced such that one void bit position exists between all data bits after transfer out operation. Therefore, all major loop functions are repeated at a 50 KHz frequency. The drawing below is arranged so that the functions may be set up in the proper synchronism. The timings are given in both field cycles and microseconds. The field cycle counts corresponds to Figure 1.



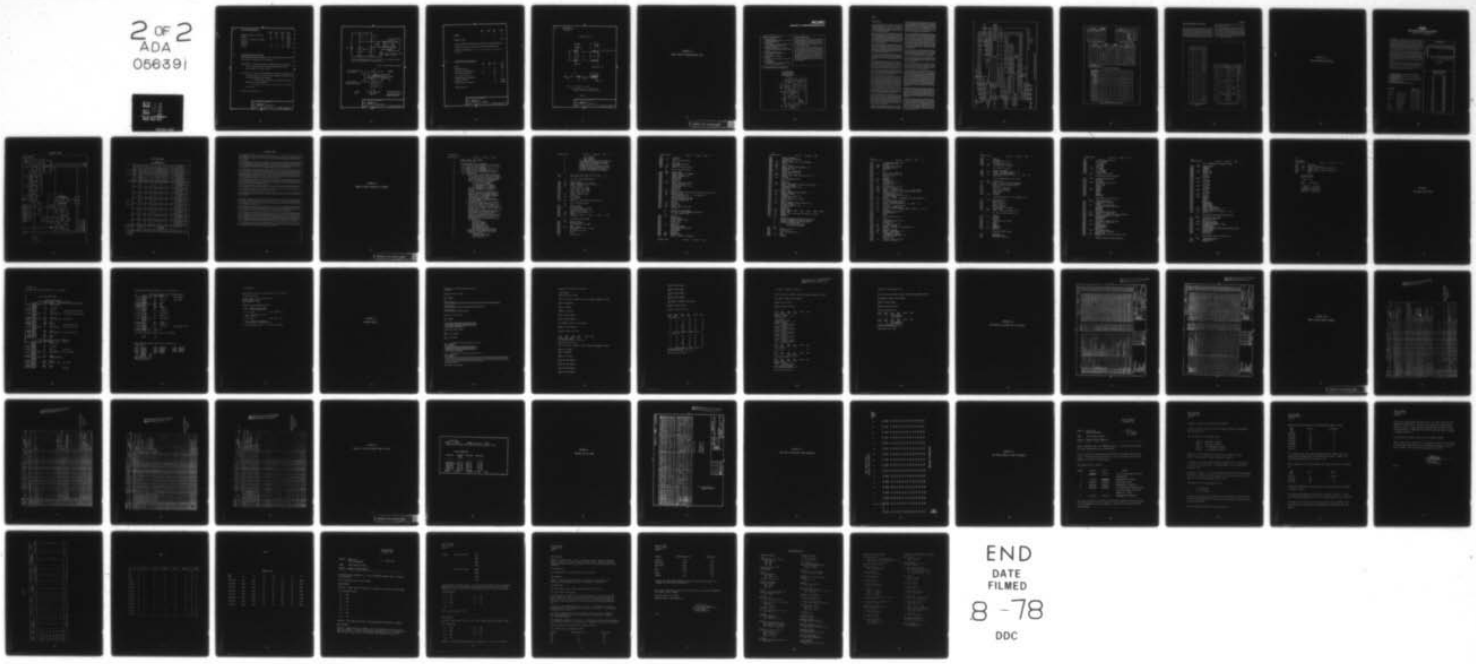
Note: t in μs, T in field cycles.
Each field cycle begins at 0° field orientation.

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV	SHEET 13 of 17

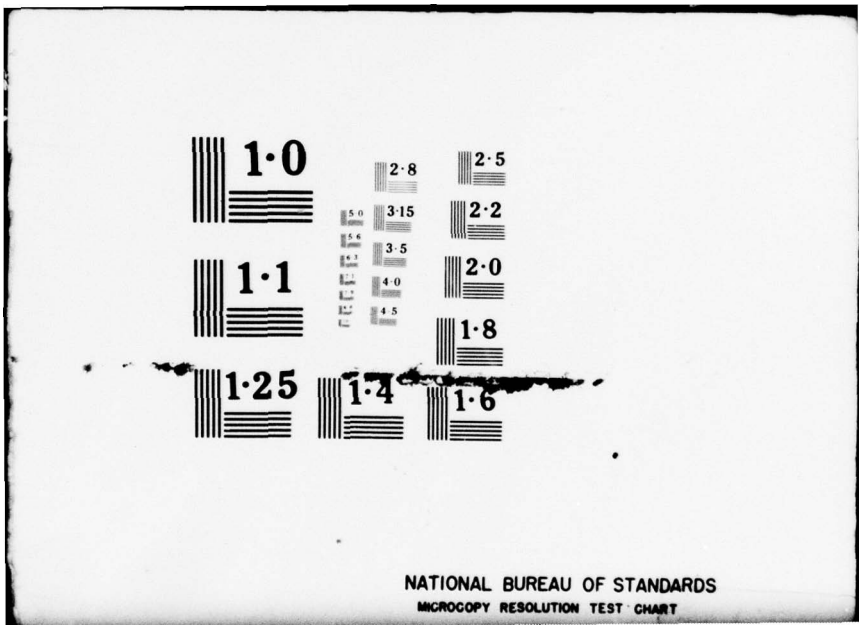
TI 7915C

AD-A056 391 DEVELCO INC SUNNYVALE CA F/G 9/2
MAGNETIC BUBBLE MEMORY SYSTEM CONCEPTS FOR FIELD DATA ACQUISITI--ETC(U)
SEP 77 L R BULDUC, T C MOORE, O L KWONG DNA001-77-C-0137
UNCLASSIFIED 1026-77930 DNA-4499F NL

2 OF 2
ADA
056391



END
DATE FILMED
8-78
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
5·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

8.0 FUNCTION RESISTANCE

	Min	Nom	Max	Unit
Detector 2 (closest to bar edge)	950	1100	1300	Ohms
Detector 1	950	1100	1300	Ohms
Detector 2 - Detector 1 mismatch	-10	-	+10	Ohms
Generator	2	3	5	Ohms
Replicator	5	6	8	Ohms
Transfer	260	310	350	Ohms

9.0 DETECTOR SIGNAL AND NOISE

The following information presumes bridge connected detectors with 5.5 ma DC flowing in each bridge arm as in Figure 4.

Signal Source: A single bubble domain passing under the two magnetoresistive detectors produces a signal of several millivolts with two positive peaks and two negative peaks.

Noise Sources: 1) Since the magnetoresistive elements are surrounded by a rotating magnetic field, any detector mismatch will produce noise at twice the rotating field frequency.

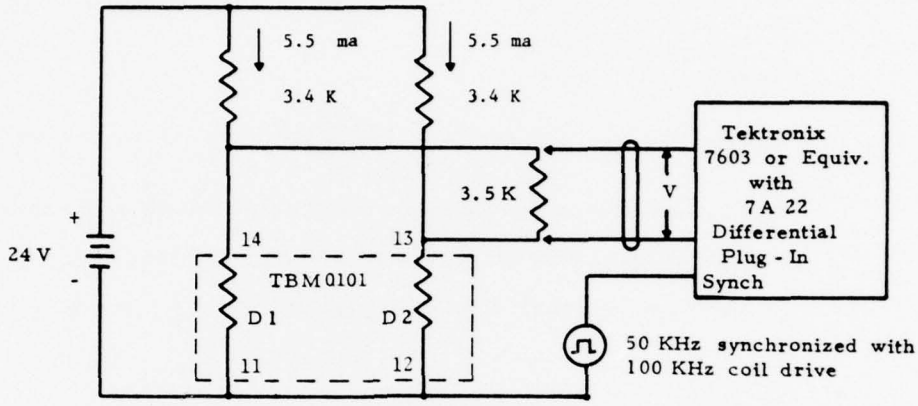
2) Control pulses can produce high frequency noise components due to circuit coupling.

3) Incorrect circuit layout can produce noise at the rotating field frequency.

See Figure 4 for test set up.

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV A	SHEET 14 of 17

TI-7915C



Read an alternating pattern of 1's and all 0's for simultaneous scope display of both bubble and no bubble signals.

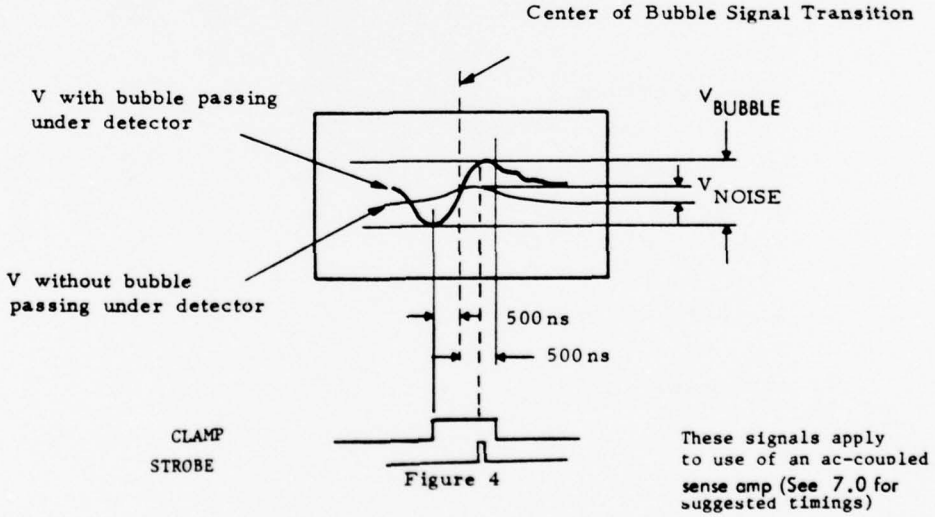


Figure 4

These signals apply to use of an ac-coupled sense amp (See 7.0 for suggested timings)

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV A	SHEET 15 of 17

TI-7915C

	Min	Nom	Max	Units
V_{BUBBLE}	2	-	7.5	mv
V_{BUBBLE} / V_{NOISE}	4	-	-	mv/mv

V_{NOISE} is defined as the maximum positive transition of the no-bubble signal within a $\pm$ 500 ns window on either side of the center of the bubble signal transition.

The bubble signal transition time shall not vary more than ± 0.1 µs under all conditions.

10.0 ENVIRONMENTAL SPECIFICATIONS

	Min	Nom	Max	Unit
Weight	-	25	-	Grams
Operating temperature	0	-	70	°C
Nonvolatile* storage temperature	-40	-	85	°C
Non-use storage temperature	-40	-	100	°C
Non-operating handling shock	-	-	300	G
Operating vibration up to 500 KHz	-	-	5	G
Humidity (Up to 70°C)	-	-	95	Percent
DC magnetic field, any direction	-	-	20	Oersteds

*Without memory loss

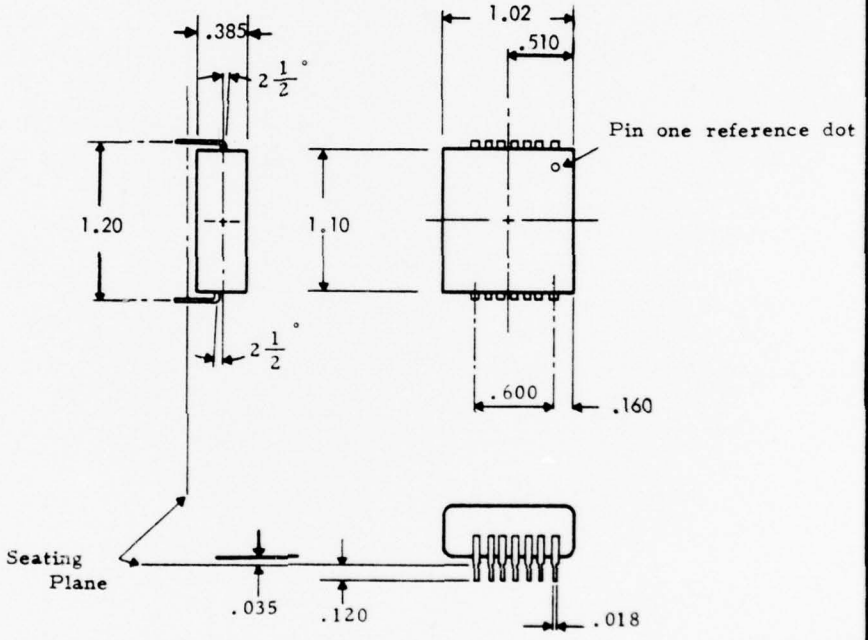
SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV	SHEET 16 of 17

TI-7915C

11.0 MECHANICAL

See Figure 5.

Mechanical Outline



Note: a) All dimensions in inches

b) Lead centerlines are located within 0.005 of true position relative to body

Figure 5

SIZE	CODE IDENT NO	DRAWING NO
A	96214	SK 2007651
SCALE	REV A	SHEET 17 of 17

TI 7015C

APPENDIX II

MC2901 FOUR-BIT MICROPROCESSOR SLICE

PRECEDING PAGE BLANK-NOT FILMED

MC2901

FOUR-BIT TTL MICROPROCESSOR SLICE

DISTINCTIVE CHARACTERISTICS

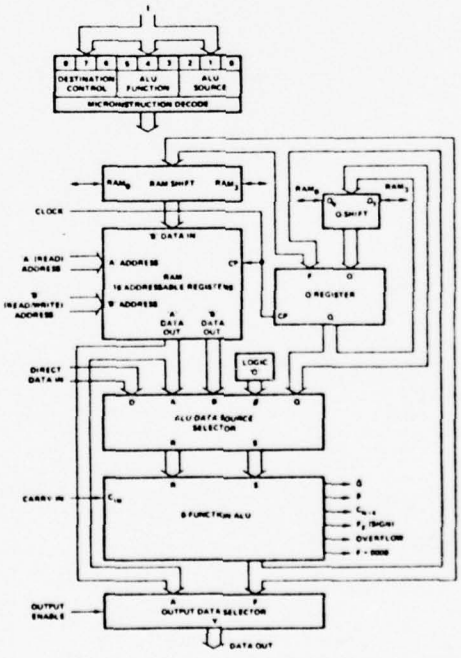
- Two-address architecture –
  Independent simultaneous access to two working registers saves machine cycles.
- Eight-function ALU –
  Performs addition, two subtraction operations, and five logic functions on two source operands.
- Flexible data source selection –
  ALU data is selected from five source ports for a total of 203 source operand pairs for every ALU function.
- Left/right shift independent of ALU –
  Add and shift operations take only one cycle.
- Four status flags –
  Carry, overflow, zero, and negative.
- Expandable –
  Connect any number of MC2901's together for longer word lengths.
- Microprogrammable –
  Three groups of three bits each for source operand, ALU function, and destination control.

GENERAL DESCRIPTION

The four-bit bipolar microprocessor slice is designed as a high-speed cascadable element intended for use in CPU's, peripheral controllers, programmable microprocessors and numerous other applications. The microinstruction flexibility of the MC2901 will allow efficient emulation of almost any digital computing machine.

The device, as shown in the block diagram below, consists of a 16-word by 4-bit two-port RAM, a high-speed ALU, and the associated shifting, decoding and multiplexing circuitry. The nine-bit microinstruction word is organized into three groups of three bits each and selects the ALU source operands, the ALU function, and the ALU destination register. The microprocessor is cascadable with full look-ahead or with ripple carry, has three-state outputs, and provides various status flag outputs from the ALU. Advanced low-power Schottky processing is used to fabricate this 40-lead LSI chip.

MICROPROCESSOR SLICE BLOCK DIAGRAM



ARCHITECTURE

A detailed block diagram of the bipolar microprogrammable microprocessor structure is shown in Figure 1. The circuit is a four-bit slice cascadable to any number of bits. Therefore, all data paths within the circuit are four bits wide. The two key elements in the Figure 1 block diagram are the 16-word by 4-bit 2-port RAM and the high-speed ALU.

Data in any of the 16 words of the Random Access Memory (RAM) can be read from the A-port of the RAM as controlled by the 4-bit A address field input. Likewise, data in any of the 16 words of the RAM as defined by the B address field input can be simultaneously read from the B-port of the RAM. The same code can be applied to the A select field and B select field in which case the identical file data will appear at both the RAM A-port and B-port outputs simultaneously.

When enabled by the RAM write enable (RAM EN), new data is always written into the file (word) defined by the B address field of the RAM. The RAM data input field is driven by a 3-input multiplexer. This configuration is used to shift the ALU output data (F) if desired. This three-input multiplexer scheme allows the data to be shifted up one bit position, shifted down one bit position, or not shifted in either direction.

The RAM A-port data outputs and RAM B-port data outputs drive separate 4-bit latches. These latches hold the RAM data while the clock input is LOW. This eliminates any possible race conditions that could occur while new data is being written into the RAM.

The high-speed Arithmetic Logic Unit (ALU) can perform three binary arithmetic and five logic operations on the two 4-bit input words R and S. The R input field is driven from a 2-input multiplexer, while the S input field is driven from a 3-input multiplexer. Both multiplexers also have an inhibit capability; that is, no data is passed. This is equivalent to a "zero" source operand.

Referring to Figure 1, the ALU R-input multiplexer has the RAM A-port and the direct data inputs (D) connected as inputs. Likewise, the ALU S-input multiplexer has the RAM A-port, the RAM B-port and the Q register connected as inputs.

This multiplexer scheme gives the capability of selecting various pairs of the A, B, D, Q and "0" inputs as source operands to the ALU. These five inputs, when taken two at a time, result in ten possible combinations of source operand pairs. These combinations include AB, AD, AQ, A0, BD, BQ, B0, DQ, D0 and Q0. It is apparent that AD, AQ and A0 are somewhat redundant with BD, BQ and B0 in that if the A address and B address are the same, the identical function results. Thus, there are only seven completely non-redundant source operand pairs for the ALU. The MC2901 microprocessor implements eight of these pairs. The microinstruction inputs used to select the ALU source operands are the I_0, I_1, and I_2 inputs. The definition of I_0, I_1, and I_2 for the eight source operand combinations are as shown in Figure 2. Also shown is the octal code for each selection.

The two source operands not fully described as yet are the D input and Q input. The D input is the four-bit wide direct data field input. This port is used to insert all data into the working registers inside the device. Likewise, this input can be used in the ALU to modify any of the internal data files. The Q register is a separate 4-bit file intended primarily for multiplication and division routines but it can also be used as an accumulator or holding register for some applications.

The ALU itself is a high-speed arithmetic/logic operator capable of performing three binary arithmetic and five logic functions. The I_3, I_4, and I_5 microinstruction inputs are used to select the ALU function. The definition of these inputs is shown in Figure 3. The octal code is also shown for reference. The normal technique for cascading the ALU of several devices is in a look-ahead carry mode. Carry generate, $\overline{G}$, and carry propagate, $\overline{P}$, are outputs of the device for use with a carry-look-ahead-generator such as the MC2902 ('182). A carry-out, C_{n+4}, is also generated and is available as an output for use as the carry flag in a status register. Both carry-in (C_n) and carry-out (C_{n+4}) are active HIGH.

The ALU has three other status-oriented outputs. These are F_3, F = 0, and overflow (OVR). The F_3 output is the most significant (sign) bit of the ALU and can be used to determine positive or negative results without enabling the three-state data outputs. F_3 is non-inverted with respect to the sign bit output Y_3. The F = 0 output is used for zero detect. It is an open-collector output and can be wire OR'ed between microprocessor slices. F = 0 is HIGH when all F outputs are LOW. The overflow output (OVR) is used to flag arithmetic operations that exceed the available two's complement number range. The overflow output (OVR) is HIGH when overflow exists. That is, when C_{n+3} and C_{n+4} are not the same polarity.

The ALU data output is routed to several destinations. It can be a data output of the device and it can also be stored in the RAM or the Q register. Eight possible combinations of ALU destination functions are available as defined by the I_6, I_7, and I_8 microinstruction inputs. These combinations are shown in Figure 4.

The four-bit data output field (Y) features three-state outputs and can be directly bus organized. An output control ($\overline{OE}$) is used to enable the three-state outputs. When $\overline{OE}$ is HIGH, the Y outputs are in the high-impedance state.

A two-input multiplexer is also used at the data output such that either the A-port of the RAM or the ALU outputs (F) are selected at the device Y outputs. This selection is controlled by the I_6, I_7, and I_8 microinstruction inputs. Refer to Figure 4 for the selected output for each microinstruction code combination.

As was discussed previously, the RAM inputs are driven from a three-input multiplexer. This allows the ALU outputs to be entered non-shifted, shifted up one position (X2) or shifted down one position (÷2). The shifter has two ports; one is labeled RAM_0 and the other is labeled RAM_3. Both of these ports consist of a buffer-driver with a three-state output and an input to the multiplexer. Thus, in the shift up mode, the RAM_3 buffer is enabled and the RAM_0 multiplexer input is enabled. Likewise, in the shift down mode, the RAM_0 buffer and RAM_3 input are enabled. In the no-shift mode, both buffers are in the high-impedance state and the multiplexer inputs are not selected. This shifter is controlled from the I_6, I_7 and I_8 microinstruction inputs as defined in Figure 4.

Similarly, the Q register is driven from a 3-input multiplexer. In the no-shift mode, the multiplexer enters the ALU data into the Q register. In either the shift-up or shift-down mode, the multiplexer selects the Q register data appropriately shifted up or down. The Q shifter also has two ports; one is labeled Q_0 and the other is Q_3. The operation of these two ports is similar to the RAM shifter and is also controlled from I_6, I_7, and I_8 as shown in Figure 4.

The clock input to the MC2901 controls the RAM, the Q register, and the A and B data latches. When enabled, data is clocked into the Q register on the LOW-to-HIGH transition of the clock. When the clock input is HIGH, the A and B latches are open and will pass whatever data is present at the RAM outputs. When the clock input is LOW, the latches are closed and will retain the last data entered. If the RAM-EN is enabled, new data will be written into the RAM file (word) defined by the B address field when the clock input is LOW.

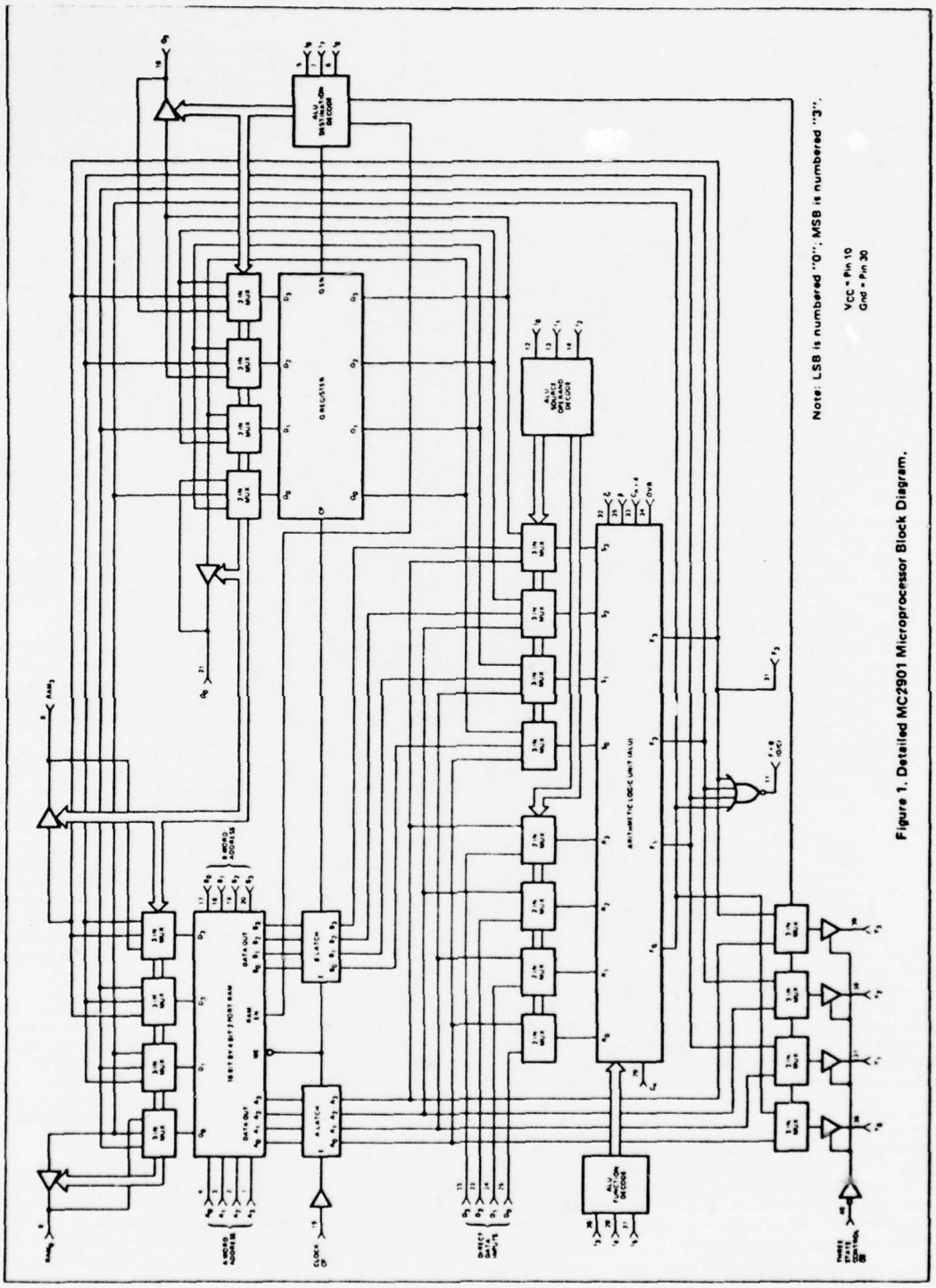


Figure 1. Detailed MC2901 Microprocessor Block Diagram.

MICRO CODE				ALU SOURCE OPERANDS	
Pin 14 I_2	Pin 13 I_1	Pin 12 I_0	Octal Code	R	S
L	L	L	0	A	Q
L	L	H	1	A	B
L	H	L	2	O	Q
L	H	H	3	O	B
H	L	L	4	O	A
H	L	H	5	D	A
H	H	L	6	D	Q
H	H	H	7	D	O

Figure 2. ALU Source Operand Control.

MICRO CODE				ALU Function	Symbol
Pin 27 I_5	Pin 28 I_4	Pin 26 I_3	Octal Code		
L	L	L	0	R Plus S	R + S
L	L	H	1	S Minus R	S − R
L	H	L	2	R Minus S	R − S
L	H	H	3	R OR S	R ∨ S
H	L	L	4	R AND S	R ∧ S
H	L	H	5	$\bar{R}$ AND S	$\bar{R}$ ∧ S
H	H	L	6	R EX-OR S	R ∀ S
H	H	H	7	R EX-NOR S	$\overline{R \forall S}$

Figure 3. ALU Function Control.

MICRO CODE				RAM FUNCTION		Q-REG. FUNCTION		Y OUTPUT	RAM SHIFTER		Q SHIFTER	
Pin 6 I_8	Pin 7 I_7	Pin 5 I_6	Octal Code	Shift	Load	Shift	Load		RAM_0	RAM_3	Q_0	Q_3
L	L	L	0	X	NONE	NONE	F → Q	F	X	X	X	X
L	L	H	1	X	NONE	X	NONE	F	X	X	X	X
L	H	L	2	NONE	F → B	X	NONE	A	X	X	X	X
L	H	H	3	NONE	F → B	X	NONE	F	X	X	X	X
H	L	L	4	DOWN	F/2 → B	DOWN	Q/2 → Q	F	F_0	IN_3	Q_0	IN_3
H	L	H	5	DOWN	F/2 → B	X	NONE	F	F_0	IN_3	Q_0	X
H	H	L	6	UP	2F → B	UP	2Q → Q	F	IN_0	F_3	IN_0	Q_3
H	H	H	7	UP	2F → B	X	NONE	F	IN_0	F_3	X	Q_3

X = Don't care. Electrically, the shift pin is a TTL input internally connected to a three-state output which is in the high-impedance state.
B = Register Addressed by B inputs.
Up is toward MSB, Down is toward LSB.

Figure 4. ALU Destination Control.

OCTAL $I_{5\,4\,3}$	I_{210} OCTAL / ALU Source / ALU Function	0	1	2	3	4	5	6	7
		A, Q	A, B	O, Q	O, B	O, A	D, A	D, Q	D, O
0	C_n = L R Plus S C_n = H	A+Q A+Q+1	A+B A+B+1	Q Q+1	B B+1	A A+1	D+A D+A+1	D+Q D+Q+1	D D+1
1	C_n = L S Minus R C_n = H	Q−A−1 Q−A	B−A−1 B−A	Q−1 Q	B−1 B	A−1 A	A−D−1 A−D	Q−D−1 Q−D	−D−1 −D
2	C_n = L R Minus S C_n = H	A−Q−1 A−Q	A−B−1 A−B	−Q−1 −Q	−B−1 −B	−A−1 −A	D−A−1 D−A	D−Q−1 D−Q	D−1 D
3	R OR S	A ∨ Q	A ∨ B	Q	B	A	D ∨ A	D ∨ Q	D
4	R AND S	A ∧ Q	A ∧ B	0	0	0	D ∧ A	D ∧ Q	0
5	$\bar{R}$ AND S	$\bar{A}$ ∧ Q	$\bar{A}$ ∧ B	Q	B	A	$\bar{D}$ ∧ A	$\bar{D}$ ∧ Q	0
6	R EX-OR S	A ∀ Q	A ∀ B	Q	B	A	D ∀ A	D ∀ Q	D
7	R EX-NOR S	$\overline{A \forall Q}$	$\overline{A \forall B}$	$\bar{Q}$	$\bar{B}$	$\bar{A}$	$\overline{D \forall A}$	$\overline{D \forall Q}$	$\bar{D}$

+ = Plus, − = Minus, ∨ = OR, ∧ = AND, ∀ = EX OR

Figure 5. Source Operand and ALU Function Matrix.

SOURCE OPERANDS AND ALU FUNCTIONS

There are eight source operand pairs available to the ALU as selected by the I_0, I_1, and I_2 instruction inputs. The ALU can perform eight functions; five logic and three arithmetic. The I_3, I_4, and I_5 instruction inputs control this function selection. The carry input, C_n, also affects the ALU results when in the arithmetic mode. The C_n input has no effect in the logic mode. When I_0 through I_5 and C_n are viewed together, the matrix of Figure 5 results. This matrix fully defines the ALU/source operand function for each state.

The ALU functions can also be examined on a "task" basis, i.e., add, subtract, AND, OR, etc. In the arithmetic mode, the carry will affect the function performed while in the logic mode, the carry will have no bearing on the ALU output. Figure 6 defines the various logic operations that the MC2901 can perform and Figure 7 shows the arithmetic functions of the device. Both carry-in LOW (C_n = 0) and carry-in HIGH (C_n = 1) are defined in these operations.

Octal I_{543}, I_{210}	Group	Function
4 0	AND	A∧Q
4 1		A∧B
4 5		D∧A
4 6		D∧Q
3 0	OR	A∨Q
3 1		A∨B
3 5		D∨A
3 6		D∨Q
6 0	EX-OR	A∀Q
6 1		A∀B
6 5		D∀A
6 6		D∀Q
7 0	EX-NOR	$\overline{A \forall Q}$
7 1		$\overline{A \forall B}$
7 5		$\overline{D \forall A}$
7 6		$\overline{D \forall Q}$
7 2	INVERT	$\overline{Q}$
7 3		$\overline{B}$
7 4		$\overline{A}$
7 7		$\overline{D}$
6 2	PASS	Q
6 3		B
6 4		A
6 7		D
3 2	PASS	Q
3 3		B
3 4		A
3 7		D
4 2	"ZERO"	0
4 3		0
4 4		0
4 7		0
5 0	MASK	$\overline{A}$∧Q
5 1		$\overline{A}$∧B
5 5		$\overline{D}$∧A
5 6		$\overline{D}$∧Q

Figure 6. ALU Logic Mode Functions. (C_n Irrelevant)

Octal	C_n = 0 (Low)		C_n = 1 (High)	
I_{543}, I_{210}	Group	Function	Group	Function
0 0	ADD	A+Q	ADD plus one	A+Q+1
0 1		A+B		A+B+1
0 5		D+A		D+A+1
0 6		D+Q		D+Q+1
0 2	PASS	Q	Increment	Q+1
0 3		B		B+1
0 4		A		A+1
0 7		D		D+1
1 2	Decrement	Q−1	PASS	Q
1 3		B−1		B
1 4		A−1		A
2 7		D−1		D
2 2	1's Comp.	−Q−1	2's Comp. (Negate)	−Q
2 3		−B−1		−B
2 4		−A−1		−A
1 7		−D−1		−D
1 0	Subtract (1's Comp)	Q−A−1	Subtract (2's Comp)	Q−A
1 1		B−A−1		B−A
1 5		A−D−1		A−D
1 6		Q−D−1		Q−D
2 0		A−Q−1		A−Q
2 1		A−B−1		A−B
2 5		D−A−1		D−A
2 6		D−Q−1		D−Q

Figure 7. ALU Arithmetic Mode Functions.

APPENDIX III

9408 MICROPROGRAM SEQUENCER

9408
MICROPROGRAM SEQUENCER
FAIRCHILD I^3L^{TM} MACROLOGIC

DESCRIPTION – The 9408 Microprogram Sequencer controls the order in which microinstructions are fetched from the control memory. It contains a 10-bit program counter, a 4-level last-in first-out stack and associated control logic. It can control up to a maximum of 1024 words of memory. For larger word capacities external paging can be used. The 9408 is controlled by a 4-bit instruction input. The instruction set includes Fetch, Conditional and Unconditional Branches, Branch to Subroutine and Return from Subroutine.

There are seven test inputs – four participate in conditional branches, and three in multiway branches. The conditional test lines are flip-flop buffered. These flip-flops can be tested individually by appropriate branch instructions. The three multiway test inputs are used to form the least significant three bits of the branch address for a multiway branch. Thus, branching occurs at one of eight unique locations depending on the bit pattern present on these three inputs.

The 9408 is designed to operate in pipeline or non-pipeline mode as desired by the user. The device operates synchronously with the clock input and can be initialized using the Master Reset input.

The 9408 is fabricated using Integrated Injection Logic (I^3L^{TM}) technology and fully compatible with all TTL families.

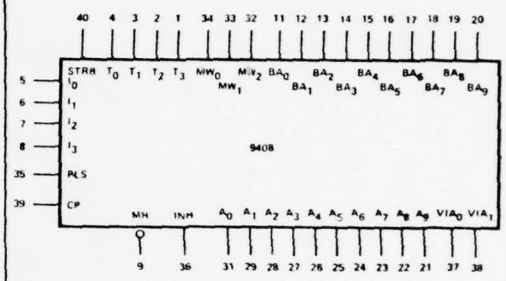
- CONTROLS 1024 WORDS OF MICROPROGRAM MEMORY (10-BIT ADDRESS)
- UNRESTRICTED BRANCHING WITHIN 10-BIT ADDRESS SPACE
- 16 INSTRUCTIONS
- FOUR FLIP-FLOP BUFFERED TEST INPUTS FOR CONDITIONAL BRANCHES
- 8-WAY BRANCH CAPABILITY
- PIPELINE/NON-PIPELINE MODE OF OPERATION

PIN NAMES

		LOADING (Note a)	
		HIGH	LOW
BA_0–BA_9	Branch Address Inputs	0.5 U.L.	0.23 U.L.
T_0–T_3	Test Inputs	0.5 U.L.	0.23 U.L.
MW_0–MW_2	Multiway Branch Inputs	0.5 U.L.	0.23 U.L.
I_1	Instruction Input	0.5 U.L.	0.23 U.L.
I_0, I_2, I_3	Instruction Inputs	1.0 U.L.	0.46 U.L.
PLS	Pipeline Select Input	0.5 U.L.	0.23 U.L.
$\overline{MR}$	Master Reset Input (Active LOW)	0.5 U.L.	0.23 U.L.
CP	Clock Pulse Input	1.0 U.L.	0.46 U.L.
STRB	Strobe Input	0.5 U.L.	0.23 U.L.
A_0–A_9	Address Outputs	10 U.L.	5.0 U.L.
VIA_0, VIA_1	VIA Outputs	10 U.L.	5.0 U.L.
$\overline{INH}$	Inhibit Output	10 U.L.	5.0 U.L.

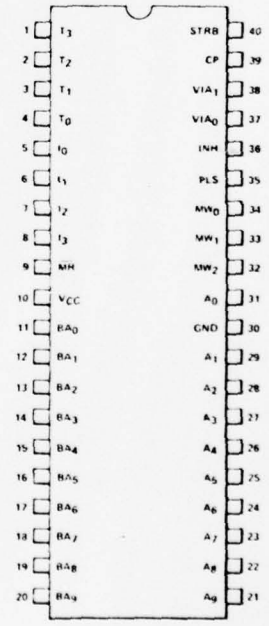
Note a: 1 TTL Unit Load (U.L.) = 40 μA HIGH, 1.6 mA LOW.

LOGIC SYMBOL



V_{CC} = Pin 10
GND = Pin 30

CONNECTION DIAGRAM
DIP (TOP VIEW)



BLOCK DIAGRAM

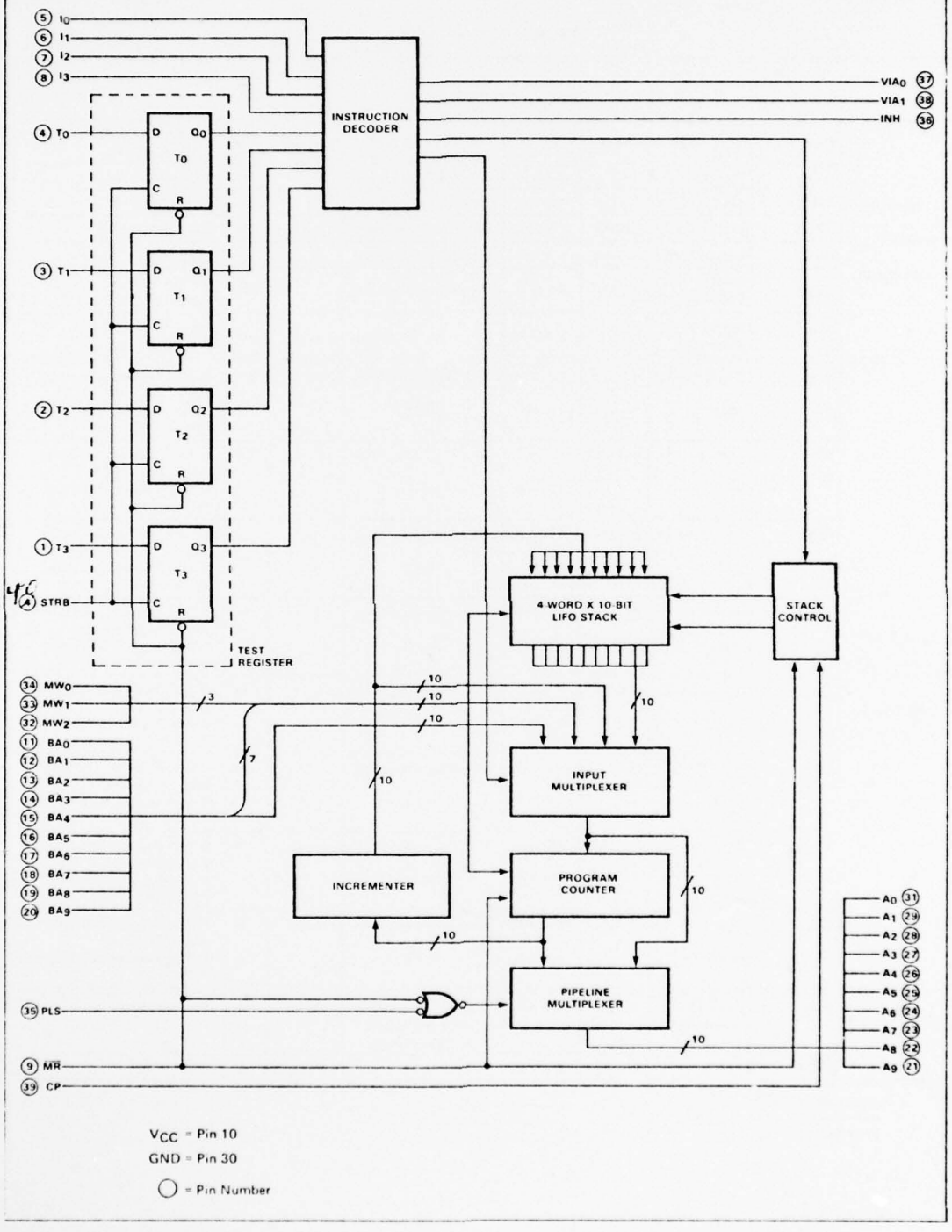


TABLE 1
9408 INSTRUCTION SET

	MNEMONIC	DEFINITION	$I_3I_2I_1I_0$	$T_3T_2T_1T_0$	$O_9O_8O_7 \cdots O_2O_1O_0$	VIA_1VIA_0	INH	DESCRIPTION OF OPERATION
Unconditional Branch Instructions	BRV_0	Branch VIA_0	L H L L	X X X X	$BA_9\ BA_8 \cdots BA_1\ BA_0$	L L	H	$BA_0 - BA_9 \rightarrow PC$
	BRV_1	Branch VIA_1	L H L H	X X X X	$BA_9\ BA_8 \cdots BA_1\ BA_0$	L H	H	$BA_0 - BA_9 \rightarrow PC$
	BRV_2	Branch VIA_2	L H H L	X X X X	$BA_9\ BA_8 \cdots BA_1\ BA_0$	H L	H	$BA_0 - BA_9 \rightarrow PC$
	BRV_3	Branch VIA_3	L H H H	X X X X	$BA_9\ BA_8 \cdots BA_1\ BA_0$	H H	H	$BA_0 - BA_9 \rightarrow PC$
	BMW	Branch Multiway	L L H H	X X X X	$BA_9\ BA_3 \cdots MW_2\ MW_0$	L L	H	$MW_0 - MW_2$, $BA_3 - BA_9 \rightarrow PC$
	BSR	Branch to Subroutine	L L L H	X X X X	$BA_9\ BA_8 \cdots BA_1\ BA_0$	L L	H	$BA_0 - BA_9 \rightarrow PC$ & Push the Stack
Conditional Branch Instructions	BTH_0	Branch on T_0 HIGH	H H L L	X X X H X X X L	$BA_9\ BA_8 \cdots BA_1\ BA_0$ PC+1	L L	H	If Test Register 0 is HIGH: $BA_0 - BA_9 \rightarrow PC$ If Test Register 0 is LOW: PC+1 → PC
	BTH_1	Branch on T_1 HIGH	H H L H	X X H X X X L X	$BA_9\ BA_8 \cdots BA_1\ BA_0$ PC+1	L L	H	If Test Register 1 is HIGH: $BA_0 - BA_9 \rightarrow PC$ If Test Register 1 is LOW: PC+1 → PC
	BTH_2	Branch on T_2 HIGH	H H H L	X H X X X L X X	$BA_9\ BA_8 \cdots BA_1\ BA_0$ PC+1	L L	H	If Test Register 2 is HIGH: $BA_0 - BA_9 \rightarrow PC$ If Test Register 2 is LOW: PC+1 → PC
	BTH_3	Branch on T_3 HIGH	H H H H	H X X X L X X X	$BA_9\ BA_8 \cdots BA_1\ BA_0$ PC+1	L L	H	If Test Register 3 is HIGH: $BA_0 - BA_9 \rightarrow PC$ If Test Register 3 is LOW: PC+1 → PC
	BTL_0	Branch on T_0 LOW	H L L L	X X X L X X X H	$BA_9\ BA_8 \cdots BA_1\ BA_0$ PC+1	L L	H	If Test Register 0 is LOW: $BA_0 - BA_9 \rightarrow PC$ If Test Register 0 is HIGH: PC+1 → PC
	BTL_1	Branch on T_1 LOW	H L L H	X X L X X X H X	$BA_9\ BA_8 \cdots BA_1\ BA_0$ PC+1	L L	H	If Test Register 1 is LOW: $BA_0 - BA_9 \rightarrow PC$ If Test Register 1 is HIGH: PC+1 → PC
	BTL_2	Branch on T_2 LOW	H L H L	X L X X X H X X	$BA_9\ BA_8 \cdots BA_1\ BA_0$ PC+1	L L	H	If Test Register 2 is LOW: $BA_0 - BA_9 \rightarrow PC$ If Test Register 2 is HIGH: PC+1 → PC
	BTL_3	Branch on T_3 LOW	H L H H	L X X X H X X X	$BA_9\ BA_8 \cdots BA_1\ BA_0$ PC+1	L L	H	If Test Register 3 is LOW: $BA_0 - BA_9 \rightarrow PC$ If Test Register 3 is HIGH: PC+1 → PC
Miscellaneous Instructions	RTS	Return from Subroutine	L L L L	X X X X	Contents of the Stack Addressed by Read Pointer	L L	L	Pop the Stack
	FTCH	FETCH	L L H L	X X X X	PC+1	L L	L	PC+1 → PC

L = LOW Level
H = HIGH Level
X = Don't Care

FUNCTIONAL DESCRIPTION – The 9408 Microprogram Sequencer, shown in the block diagram consists of a 10-bit Program Counter (PC), a 4-word by 10-bit Last-In First-Out (LIFO) Stack with associated control, an Input Multiplexer, a Pipeline Multiplexer, an Instruction Decoder, a 10-bit Incrementer, and a 4-bit Test Register comprised of four edge-triggered D flip-flops.

The Pipeline Multiplexer has two ports – the PC output provides the input port for the non-pipeline mode and the Input Multiplexer output provides the input port for the pipeline mode. Port selection is controlled by the Pipeline Select (PLS) and Master Reset ($\overline{MR}$) inputs. A LOW level on the $\overline{MR}$ input forces the non-pipeline mode of operation and clears the PC. Thus when the 9408 is initialized by the $\overline{MR}$ input the A_0 through A_9 outputs are LOW regardless of the state of the PLS input. A LOW level on the PLS input specifies non-pipeline mode and a HIGH specifies pipeline mode.

The Program Counter is a 10-bit edge-triggered register. The LOW-to-HIGH transition on the Clock (CP) input loads the Input Multiplexer output into the PC. Because of the edge-triggered nature of the PC register, the PC output remains static for a full clock cycle. Thus, in the non-pipeline mode, the PC output can be used to address a control memory built with static devices without storing the memory output in an external microinstruction register. However, in the pipeline mode, the 9408 provides the next address information as soon as available; therefore, execution of a microinstruction can be overlapped with the fetching of the next microinstruction. To ensure microinstruction stability for a full clock cycle, the control-memory output should be buffered with an external microinstruction register.

The Input Multiplexer receives data from four different sources. One port is the output of the LIFO Stack, a second is the output of the 10-bit Incrementer. The Incrementer always adds one to the PC contents. The third and fourth ports are the branch and multiway-branch ports, the latter comprised of the seven most significant Branch Address inputs (BA_3 through BA_9) and the three Multiway inputs, (MW_0 through MW_2).

The 4-word by 10-bit LIFO Stack is a RAM and receives data from the Incrementer output. The stack control logic generates the appropriate control signals, while stack pointers in the Stack Control generate the read and write addresses.

The 4-bit Test Register consists of four type-D flip-flops. The data inputs, which are the four Test inputs (T_0 through T_3), are loaded on the LOW-to-HIGH transition of the Strobe input (STRB).

The Instruction Decoder receives the 4-bit Instruction input (I_0 through I_3) and the Test Register output and generates the VIA_0, VIA_1 and Inhibit (INH) outputs of the 9408. In addition, it generates appropriate logic signals for the Stack Control and Input Multiplexer.

Stack Control – The 9408 has a 4-level subroutine nesting capability as detailed in *Figure 1*. The R_0 and R_1 (Read Address) inputs to the 4-word by 10-bit LIFO Stack specify the address from which information will be read. The W_0 and W_1 (Write Address) inputs specify the address into which information will be written, and the 9408 Incrementer output provides the information to be written into the stack (see block diagram). In addition, writing into the memory is controlled by the Write Enable ($\overline{WE}$) and $\overline{CP}$ inputs.

The R_0, R_1 and W_0, W_1 inputs of the LIFO Stack are derived from the outputs of a 3-bit edge-triggered register called the Stack Pointer (SP). The least significant two bits (SP_0 and SP_1) of this register are the read address inputs to the memory. The SP outputs are also connected to a Stack-Pointer Incrementer and a Decrementer that generate SP + 1 and SP – 1 respectively. The least significant two bits of the Incrementer are the write address bits for the memory.

The outputs of the Incrementer, Decrementer and the Stack Pointer are fed as inputs to a 3-port Stack-Pointer Multiplexer which, in turn, feeds the Stack Pointer inputs. Stack pointer loading always occurs on the LOW-to-HIGH transition of the $\overline{CP}$ input. The $\overline{MR}$ input clears the Stack Pointer. The Stack Pointer Control receives two inputs from the 9408 Instruction Decoder – the BSR input, which is active whenever a Branch-to-Subroutine (BSR) instruction is present on the I_0 through I_3 inputs, and the RTS input, which is active whenever a Return-from-Subroutine (RTS) instruction is specified. The port selection of the Stack Pointer Multiplexer is controlled by the outputs of the Stack Pointer Control. For all 9408 instructions except BSR and RTS, the Stack Pointer Multiplexer selects the Stack Pointer outputs as the instruction source.

Writing into the memory takes place whenever the $\overline{WE}$ and $\overline{CP}$ inputs are LOW. Note that the most significant register bit, SP_2, controls the $\overline{WE}$ input to prevent writing into the memory when all four locations are filled with return addresses. Thus the 9408 does not store and return addresses beyond four nesting levels.

APPENDIX IV

MAGNETIC BUBBLE FORTRAN TEST PROGRAM

PRECEDING PAGE BLANK-NOT FILMED

```
KB:<MBMV08.LST

FORTRAN V09.02                  09:08:36    04-AUG-77   PAGE    1

C       MAGNETIC BUBBLE TEST PROGRAM
C       L.R.BULDUC JULY 1977
C
C       THIS PROGRAM HAS 4 MAIN FUNCTIONS
C       (1)WRITE-WRITES 144 ASCII CHARACTERS FROM KEYBOARD
C             INTO THE FILE SPECIFIED
C       (2)READ-READS 144 ASCII CHARACTERS FROM THE FILE
C            SPECIFIED TO THE KEYBOARD
C       (3)DUMP-READS 144 ASCII CHARACTERS FROM EACH FILE
C             IN BLOCK SPECIFIED AND OUTPUTS ON KB
C       (4)TEST-THREE MODES
C               (A)FILE INTEGRITY TEST-WRITES 4 REPETITIVE
C                  BYTES  INTO EACH FILE,EACH FILE HAVING
C                  A DIFFERENT PATTERN WITH THE FILE NUMBER
C                  BEING THE BASE.IN THE FIRST WRITE PASS
C                  AND ALL SUCCEEDING ODD NUMBERED WRITE
C                  PASSES THE PATTERN IS AS FOLLOWS
C                       HI BYTE OF FILE ADDRESS
C                       LO BYTE OF FILE ADDRESS
C                       1'S COMPLEMENT OF HI ADDRESS
C                       1'S COMPLEMENT OF LO ADDRESS
C                  ALL EVEN NUMBERED WRITE PASSES HAVE
C                  THE 1'S COMPLEMENT OF THE ABOVE
C                  PATTERN
C                  ALL FILES ARE FIRST WRITTEN AND THEN
C                  READ IN SUCCESSION.THE READ PROCESS IS
C                  REPEATED FOR THE DESIGNATED NUMBER OF
C                  TIMES UNLESS THE MAX NUMBER OF ERRORS
C                  SPECFIED/FILE IS EXCEEDED.
C                  THE WRITE PROCESS IS REPEATED THE
C                  SPECIFIED NUMBER OF TIMES COMPLEMENTING
C                  THE PATTERN EACH TIME
C               (B)PATTERN TEST-TESTS EACH FILE SPECIFIED
C                  INDIVIDUALLY BEFORE PROCEEDING TO THE
C                  NEXT FILE.PATTERNS OF 10 DIFFERENT BYTE
C                  VALUES MAY BE SPECIFIED.PATTERNS ARE TO
C                  BE SPECIFIED AS DECIMAL VALUES BETWEEN THE
C                  RANGE OF -128 TO +127
C                  THE PATTERN WILL BE WRITTEN INTO THE FILE
C                  AND READ BACK THE SPECIFIED NUMBER OF
C                  TIMES UNLESS THE MAX ERROR COUNT
C                  PER FILE IS EXCEEDED THE PROCESS WILL BE
C                  REPEATED FOR THE NUMBER OF WRITE PASSES
C                  BEFORE PROCEEDING TO THE NEXT FILE
C                  TEST PROGRAMS WILL PRINT THE FOLLOWING
C                  INFORMATION-
C                       FILE NUMBER
C                       WRITE PASS NUMBER
C                       READ PASS NUMBER
C                       DATA WRITTEN IN FILE
C                       DATA READ FROM FILE
C                       BOARD NUMBER IN ERROR
C                       WORD NUMBER ERROR WAS DETECTED
C               (C)GENERATE PATTERN-COMBINATION OF A&B
C                  ABOVE.DOES SIX DIFFERENT VARIATIONS
C                  OF SPECIFY PATERN-
C                       1-WORD0=0,WORD1=-1,ETC
C                       2-WORD0=-1,WORD1=0,ETC.
C                       3-ALL WORDS=-86
```

```
C                       4-ALL WORDS=85
C                       5-ALL WORDS=0
C                       5-ALL WORDS=-1
C                   THE ABOVE TESTS EVALUATE ALL CONDITIONS
C                   OF ALTERNATING BUBBLES IN MAJOR AND
C                   MINOR LOOPS,ALL OR ABSENCE OF BUBBLES
C                   IN MAJOR AND MINOR LOOPS.
C                   PATTERNS 7&8 ARE THE TWO ADDRESS BASED
C                   PATTERNS OF THE FILE INTEGRITY TESTS.
C                   PATTERNS ARE REPEATED MOD 8 UNTIL THE
C                   SPECIFIED WRITE PASSES IS EXCEEDED
C
C
C
0001            BYTE A(145),B(80),C(10),W,R,T,D,F,S,G
0002            DATA W,R,T,D,F,S,G/'W','R','T','D','F','S','G'/
C
C       INTITIAL COMMAND DETERMINATION
C
0003    100     WRITE (6,800)
0004            READ(6,801)NCNT,(B(J),J=1,NCNT)
0005    800     FORMAT ('0READ,WRITE,TEST OR DUMP ',/,' ')
0006    801     FORMAT (Q,80A1)
0007            IF (B(1).EQ.W) GO TO300
0008            IF(B(1).EQ.R) GO TO 200
0009            IF(B(1).EQ.T) GO TO 2000
0010            IF(B(1).EQ.D) GO TO 600
0011            WRITE (6,900)
0012    900     FORMAT (' WRONG ANSWER')
0013            GO TO 100
0014    400     WRITE (6,900)
0015            WRITE(6,901)
0016            GO TO 100
0017    901     FORMAT (' LIMIT ADDRESS 0-640 INCLUSIVE',/,' ')
C
C       ASCII FILE READ ROUTINE
C
0018    200     WRITE (6,810)
0019            READ (6,811)NUM
0020            IF (NUM .GE. 641) GO TO 400
0021            CALL MACRED (NUM,A(1))
0022            WRITE (6,812) (A(J),J=1,144)
0023    810     FORMAT (' FILE NUMBER ',/,' ')
0024    811     FORMAT (I5)
0025    812     FORMAT (' ',40A1,/,' ',40A1,/,' ',40A1,/,' ',40A1)
0026            GO TO 100
C
C       ASCII FILE WRITE ROUTINE
C
0027    300     WRITE (6,810)
0028            READ (6,811)NUM
0029            IF (NUM.GE.641) GO TO 400
0030            I=0
0031    350     WRITE (6,820)
0032            READ(6,801)NCNT,(B(J),J=1,NCNT)
0033            I1=I+NCNT
0034            IF (I1.GT.144)I1=144
0035            I2=1+I
0036            I3=1
0037            DO 310 J=I2,I1
```

```
0038            A(J)=B(I3)
0039    310     I3=I3+1
0040            IF (I1.EQ.144)GO TO320
0041            I=I1
0042            GO TO 350
0043    320     CALL MACWRT (NUM,A(1))
0044            GO TO 100
0045    820     FORMAT (' WRITE IN DATA'/)
        C
        C       ASCII FILE DUMP ROUTINE
        C
0046    600     WRITE(6,601)
0047    601     FORMAT(' BEGIN FILE NUMBER?',/,' ')
0048            READ(6,705)NUM
0049            IF(NUM.LE.640)GOTO 610
0050            WRITE(6,900)
0051            WRITE(6,901)
0052            GOTO 600
0053    610     WRITE(6,611)
0054    611     FORMAT(' END FILE NUMBER?',/,' ')
0055            READ(6,705)NUMMAX
0056            IF(NUMMAX.LE.640)GOTO 620
0057            WRITE(6,900)
0058            WRITE(6,901)
0059            GOTO 610
0060    620     WRITE(6,621)
0061    621     FORMAT('0',T10,'**************************************')
0062            WRITE(6,622)NUM
0063    622     FORMAT(' MBM FILE DUMP',/,' FILE NUMBER ',I3)
0064            CALL MACRED(NUM,A(1))
0065            WRITE(6,812).(A(J),J=1,144)
0066            IF(NUM.EQ.NUMMAX) GOTO 690
0067            IF(NUM.NE.640)GOTO 630
0068            NUM=-1
0069    630     NUM=NUM+1
0070            GOTO 620
0071    690     WRITE(6,691)
0072    691     FORMAT(' END OF FILE DUMP')
0073            GOTO 100
        C
        C       PATTERN TEST STARTS HERE
        C       SET KNTRL TO 0 TO DETERMINE PATTERN AND
        C       NOT FILE INTEGRITY TEST
        C
0074    700     KNTRL=0
0075            WRITE(6,601)
0076            READ(6,705)NUM
0077            IF(NUM.LE.640) GOTO707
0078            WRITE(6,900)
0079            WRITE(6,901)
0080            GOTO 700
0081    707     WRITE(6,611)
0082            READ(6,705)NUMMAX
0083            IF(NUMMAX.LE.640)GOTO 708
0084            WRITE(6,900)
0085            WRITE(6,901)
0086            GOTO 707
0087    708     NUMBGN=NUM
0088    500     WRITE(6,501)
0089    501     FORMAT(' NUMBER OF PATTERNS?',/,' ')
```

```
  /L
FORTRAN V09.02                    09:08:36    04-AUG-77   PAGE    4

0090            READ(6,705)NUMPAT
0091            IF(NUMPAT.LE.10)GOTO510
0092            WRITE(6,503)
0093    503     FORMAT(' PLEASE LIMIT TO 10 PATTERNS!!!')
0094            GOTO 500
0095    510     K=1
0096    513     WRITE(6,511)K
0097    511     FORMAT(' ENTER PATTERN NUMBER ',I2,/,' ')
0098            READ(6,512) IPAT
0099    512     FORMAT(I4)
0100            IF(IPAT.LT.-128)GOTO 515
0101            IF(IPAT.LE.127)GOTO 516
0102    515     WRITE(6,514)
0103    514     FORMAT(' LIMIT RANGE FROM -128 TO +127 !!')
0104            GOTO 513
0105    516     C(K)=IPAT
0106            IF(K.EQ.NUMPAT)GOTO 517
0107            K=K+1
0108            GOTO 513
0109    517     WRITE(6,518)
0110    518     FORMAT(' MAX.NUMBER OF ERRORS TO BE PRINTED?',/,' ')
0111            READ(6,705)MAXERR
0112            IF(MAXERR.LE.100)GOTO 709
0113            WRITE(6,706)
0114            GOTO 517
0115    709     WRITE(6,704)
0116    704     FORMAT(' NUMBER OF WRITE PASSES?',/,' ')
0117            READ(6,705)NUMWRT
0118    705     FORMAT(I5)
0119            IF(NUMWRT.LE.100)GOTO 710
0120            WRITE(6,706)
0121    706     FORMAT(' PLEASE LIMIT TO 100')
0122            GOTO 709
0123    710     WRITE(6,711)
0124    711     FORMAT(' NUMBER OF READS PER WRITE?',/,' ')
0125            READ(6,705)NUMRD
0126            IF(NUMRD.LE.100)GOTO 720
0127            WRITE(6,706)
0128            GOTO 710
0129    720     WRITE(6,702)
0130    702     FORMAT('0WRITE   READ    GOOD    BAD     BOARD   WORD')
0131            WRITE(6,703)
0132    703     FORMAT(' PASS',T10,'PASS',T18,'DATA',T26,'DATA')
0133            KERRAC=0
0134    799     KNTWRT=1
        C
        C       COMMON ROUTINES FOR PATTERN AND FILE FOLLOW
        C       REQUIRED INFORMATION SET UP BY FILE TEST
        C       ROUTINE TO CONTROL FOLLOWING ROUTINES
        C       KNTRL=0 FOR PATTERN TEST
        C               =-1 FOR FILE TEST WRITE
        C               =+1 FOR FILE TEST READ
        C
0135    714     K=1
0136    715     DO 721 J=1,144,1
0137            A(J)=C(K)
0138            IF(K.LT.NUMPAT)GOTO 721
0139            K=0
0140    721     K=K+1
0141            KNTERR=0
```

```
*N/L
FORTRAN V09.02                    09:08:36    04-AUG-77    PAGE     5

0142              IF(NUMWRT.EQ.0) GOTO 1000
0143              CALL MACWRT(NUM,A(1))
0144     1000     KNTRD=1
0145     1001     IF(NUMRD.EQ.0) GOTO 770
0146     730      CALL MACRED(NUM,A(1))
0147              NUMWRD=0
0148              J=1
0149              K=1
0150     738      IF(A(J).EQ.C(K))GOTO 735
0151              IF(A(J).GT.C(K))GOTO 731
0152              MDIFF=C(K)-A(J)
0153              GOTO 732
0154     731      MDIFF=A(J)-C(K)
0155     732      NUMBRD=0
0156     733      MDIFF=MDIFF/2
0157              IF (MDIFF) 734,736,734
0158     734      NUMBRD=NUMBRD+1
0159              GOTO 733
0160     736      IF(KNTERR)3810,3820,3810
0161     3820     WRITE(6,701)NUM
0162     701      FORMAT(' FILE NUMBER ',I3)
0163     3810     WRITE(6,739)KNTWRT,KNTRD,C(K),A(J),NUMBRD,NUMWRD
0164     739      FORMAT(' ',I4,T10,I4,T18,I4,T26,I4,T34,I3,T42,I4)
0165              KNTERR=KNTERR+1
0166              IF(KNTERR.NE.10)GOTO 540
0167              IF(KNTERR-NUMWRD+1) 535,729,535
0168     729      WRITE(6,531)KNTWRT
0169     531      FORMAT(' ',I4,T10,'****APPARENT FATAL WRITE ERROR****')
0170              GOTO 745
0171     535      IF(KNTERR.NE.KNTRD) GOTO 540
0172              WRITE(6,536)NUMBRD,NUMWRD
0173     536      FORMAT(' WRITE FAILURE-BOARD ',I3,',WORD ',I3)
0174              GOTO 745
0175     540      IF(KNTERR.LT.MAXERR)GOTO 735
0176              WRITE(6,541)KNTWRT,MAXERR
0177     541      FORMAT(' ',I4,T10,'****MAX.NUMBER OF ERRORS(',I3,')****')
0178              GOTO 745
0179     735      IF(NUMWRD.EQ.143)GOTO 785
0180              NUMWRD=NUMWRD+1
0181              J=J+1
0182              K=K+1
0183              IF(K.LE.NUMPAT)GOTO 738
0184              K=1
0185              GOTO 738
0186     785      IF(KNTRD.GE.NUMRD)GOTO 751
0187              KNTRD=KNTRD+1
0188              GOTO 730
0189     751      IF (KNTERR)740,745,740
0190     741      WRITE(6,742)KNTWRT
0191     742      FORMAT(' ',I4,T10,'*****NO ERRORS****')
0192              GOTO 745
0193     740      WRITE(6,743)KNTERR
0194     743      FORMAT(' TOTAL ERRORS ',I5)
0195     745      KERRAC=KERRAC+KNTERR
0196              IF(NUMWRT.EQ.0) GOTO770
0197              IF(KNTWRT.GE.NUMWRT)GOTO 770
0198              KNTWRT=KNTWRT+1
0199              GOTO 714
0200     770      IF(NUM.EQ.NUMMAX) GOTO 760
0201              IF(NUM.LT.640)GOTO 777
```

```
*N/L
FORTRAN V09.02                    09:08:36    04-AUG-77    PAGE     6

0202                NUM=-1
0203     777        NUM=NUM+1
0204                IF(KNTRL0.EQ.0)GOTO799
0205                GOTO 714
0206     760        IF(KNTRL)3200,762,3600
0207     762        IF(KNTRL0.NE.0) GOTO 9000
                    C
                    C       EXIT HERE IF FILE TEST
                    C
0208                WRITE(6,780)KERRAC
0209     780        FORMAT(' ACCUMULATED ERRORS=',I5)
0210                WRITE(6,761)NUMBGN,NUMMAX
0211     761        FORMAT(' END OF TEST ON FILES ',I3,' THRU ',I3)
0212                GO TO 100
                    C
                    C       INITIAL DETERMINATION OF TEST TYPE
                    C
0213     2000       WRITE(6,2001)
0214     2001       FORMAT(' TYPE OF TEST-FILE INTEGRITY,
                   2SPECIFY PATTERN OR GENERATE PATTERN?',/,' ')
0215                READ(6,801)NCNT,(B(J),J=1,NCNT)
0216                KNTRL0=0
0217                IF(B(1).EQ.S)GOTO 700
0218                IF(B(1).EQ.F)GOTO2010
0219                IF(B(1).EQ.G)GOTO 2009
0220                WRITE(6,900)
0221                GOTO 2000
0222     2009       KNTRL0=1
                    C
                    C       DETERMINE PARAMETERS OF FILE INTEGRITY TEST
                    C
0223     2010       WRITE(6,518)
0224                READ(6,705)MAXERR
0225                WRITE(6,704)
0226                READ(6,705)NUMFWT
0227                WRITE(6,711)
0228                READ(6,705)NUMFRD
0229                IF(KNTRL0.NE.0) GOTO 5000
0230     3002       WRITE(6,3001)
0231     3001       FORMAT(' MBM FILE INTEGRITY TEST')
                    C
                    C       SET INITIAL CONDITIONS FOR TEST
                    C
0232                KNTFWT=1
0233     3005       KNTERF=0
0234                KNTRLP=0
0235                KNTWRT=1
0236                IF(NUMFWT.EQ.0)GOTO 3500
0237     3220       KNTRL=-1
0238                NUM=0
0239                NUMWRT=1
0240                NUMRD=0
0241     3300       NUMMAX=NUM
0242                NUMPAT=4
                    C
                    C       DETERMINE PATTERN VALUES
                    C
0243                NUMLO=NUM
0244                NUMHI=NUMLO/256
0245                NUMLO=NUMLO-NUMHI*256

*
```

```
0246            IF(KNTRLP.EQ.0)GOTO 3310
0247            C(1)=-1-NUMHI
0248            C(2)=-1-NUMLO
0249            C(3)=NUMHI
0250            C(4)=NUMLO
0251            GOTO 3320
0252    3310    C(1)=NUMHI
0253            C(2)=NUMLO
0254            C(3)=-1-NUMHI
0255            C(4)=-1-NUMLO
0256    3320    IF(KNTRL)714,714,1001
        C
        C       RETURN HERE FROM WRITE PROCESS
        C
0257    3200    IF(NUM.EQ.640)GOTO 3500
0258            NUM=NUM+1
0259            GOTO 3300
0260    3500    IF(NUMFRD.EQ.0)GOTO 4100
0261            WRITE(6,702)
0262            WRITE(6,703)
0263            KNTFRD=1
0264            KNTRD=1
0265    3550    NUMWRT=0
0266            NUMRD=1
0267            NUM=0
0268            KNTRL=1
0269            KERRFF=0
0270            GOTO 3300
        C
        C       RETURN HERE FROM READ PROCESS
        C
0271    3600    KNTERF=KNTERF+KNTERR
0272            IF(KNTERR.EQ.0)GOTO3601
0273            KERRFF=1
0274    3601    IF(NUM.EQ.640) GOTO 4000
0275            IF(KNTERR.EQ.MAXERR)GOTO 4100
0276            KNTERR=0
0277            NUM=NUM+1
0278            GOTO 3300
0279    4000    IF(KERRFF.NE.0)GOTO 4010
0280            WRITE(6,4011)KNTFWT,KNTFRD
0281    4011    FORMAT(' ',I4,T10,I4,T18,'***NO ERRORS***')
0282    4010    IF(KNTFRD.EQ.NUMFRD) GOTO 4100
0283            KNTFRD=KNTFRD+1
0284            KNTRD=KNTFRD
0285            GOTO 3550
0286    4100    IF(KNTFWT.EQ.NUMFWT)GOTO 4200
0287            IF(KNTRLP.EQ.0)GOTO 4120
0288            KNTRLP=-1
0289    4120    KNTRLP=KNTRLP+1
0290            KNTFWT=KNTFWT+1
0291            KNTWRT=KNTFWT
0292            GOTO 3220
0293    4200    WRITE(6,780)KNTERF
0294            IF(KNTRL0.NE.0)GOTO9900
0295            WRITE(6,4201)
0296    4201    FORMAT(' END OF MBM FILE INTEGRITY TEST')
0297            GOTO 100
        C
        C       GENERATE PATTERN ROUTINE STARTS HERE
```

```
        C       SET UP PARAMETERS FOR COMMON ROUTINES
        C
0298    5000    NUMGWT=NUMFWT
0299            NUMFWT=2
0300            KNTERG=0
0301            KNTGWT=1
0302    5001    WRITE(6,702)
0303            WRITE(6,703)
0304            C(1)=0
0305            C(2)=-1
0306            GOTO 6000
0307    5002    C(1)=-1
0308            C(2)=0
0309            GOTO 6000
0310    5003    C(1)=-86
0311            C(2)=-86
0312            GOTO 6000
0313    5004    C(1)=85
0314            C(2)=85
0315            GOTO 6000
0316    5005    C(1)=0
0317            C(2)=0
0318            GOTO 6000
0319    5006    C(1)=-1
0320            C(2)=-1
0321    6000    NUMPAT=2
0322            KERRAC=0
0323            KNTRL=0
0324            NUM=0
0325            NUMMAX=640
0326            NUMRD=NUMFRD
0327            KNTWRT=KNTGWT
0328            NUMWRT=KNTWRT
0329            WRITE(6,6001)KNTRL0
0330    6001    FORMAT(' PATTERN NUMBER ',I1)
0331            WRITE(6,6002)C(1),C(2)
0332    6002    FORMAT(' WORD0=',I4,',WORD1=',I4,',ETC.')
0333            GOTO 714
        C
        C       RETURN HERE FROM COMMON PATTERN ROUTINE
        C
0334    9000    IF(KERRAC.NE.0)GOTO9005
0335            WRITE(6,742)KNTGWT
0336            GOTO 9001
0337    9005    WRITE(6,780)KERRAC
0338            KNTERG=KNTERG+KERRAC
0339    9001    IF(NUMGWT.EQ.KNTGWT)GOTO 8000
0340            KNTGWT=KNTGWT+1
0341            KNTRL0=KNTRL0+1
0342            GOTO(5001,5002,5003,5004,5005,5006,5007)KNTRL0
0343    5007    KNTFWT=KNTGWT
0344            NUMFWT=KNTFWT+1
0345            WRITE(6,3001)
0346            GOTO 3005
        C
        C       RETURN HERE FROM FILE INTEGRITY TEST
        C
0347    9900    KNTERG=KNTERG+KNTERF
0348            KNTGWT=KNTGWT+1
0349            KNTRL0=0
```

```
*N/L

W311 000000
FORTRAN V09.02                    09:08:36    04-AUG-77    PAGE     9

0350              GOTO 9001
0351     8000     WRITE(6,8001)KNTERG
0352     8001     FORMAT(' TOTAL ACCUMULATED ERROR=',I5)
0353              WRITE(6,8011)
0354     8011     FORMAT(' END OF GENERATE PATTERN TEST')
0355              GOTO 100
0356              END

         ROUTINES CALLED:
         MACRED, MACWRT

         OPTIONS =/OP:1

         BLOCK        LENGTH
         MAIN.    3873    (017102)*

         **COMPILER ----- CORE**
            PHASE       USED  FREE
         DECLARATIVES 00783 02280
         EXECUTABLES  01183 01880
         ASSEMBLY     02750 05020
*
```

APPENDIX V

MBM SUBROUTINES, MACRO

```
 KB:<MBM1.LST

MBM SUBROUTINES MACRO V06-04A 07-JUL-77 15:09 PAGE 1


1                       ;
2                       ;
3                       .TITLE  MBM SUBROUTINES
4                       ;
5                       ;
6                               GLOBL   MACWRT,MACRED
7                       ;SUBROUTINE TO WRITE 144. BYTES INTO MBM
8               011000          MASKW=011000    ;WRITE MASK FOR UPPER ADDRESS BITS
9  000000 012767 MACWRT: MOV     #MASKW,MASK1
                011000
                000210
10 00006 004467         JSR     R4,ADDXMT
                000122
11 00012 012701         MOV     #60.,R1         ;COUNT FOR FIRST 60 BYTES
                000074
12 00016 032737 LOOP3:  BIT     #1000,@#167774  ;TEST READY TO RECIEVE DATA
                001000
                167774
13 00024 001774         BEQ     LOOP3
14 00026 112037 LOOP4:  MOVB    (R0)+,@#167772  ;MOVE A BYTE
                167772
15 00032 005301         DEC     R1
16 00034 001374         BNE     LOOP4
17 00036 012701         MOV     #84.,R1         ;COUNT REST OF DATA
                000124
18 00042 032737 LOOP5:  BIT     #200,@#167770   ;TEST FOR FIFO FULL
                000200
                167770
19 00050 001374         BNE     LOOP5
20 00052 112037         MOVB    (R0)+,@#167772  ;NOT FULL MOVE BYTE
                167772
21 00056 005301         DEC     R1
22 00060 001370         BNE     LOOP5
23 00062 000205         RTS     R5
24
25                      ;SUBROUTINE TO READ 144. BYTES FROM MBM
26              010400          MASKR=010400            ;READ MASK
27 00064 012767 MACRED: MOV     #MASKR,MASK1
                010400
                000124
28 00072 004467         JSR     R4,ADDXMT
                000036
29 00076 012701         MOV     #144.,R1        ;BYTE COUNT
                000220
30 00102 032737 LOOP6:  BIT     #1000,@#167774  ;TEST FOR READY
                001000
                167774
31 00110 001774         BEQ     LOOP6
32 00112 032737 LOOP7:  BIT     #100000,@#167770
                100000
                167770
33 00120 001374         BNE     LOOP7
34 00122 113720         MOVB    @#167774,(R0)+  ;GET A BYTE
                167774
35 00126 005301         DEC     R1
36 00130 001370         BNE     LOOP7
37 00132 000205         RTS     R5              ;FINISHED
```

```
MBM SUBROUTINES MACRO V06-04A 07-JUL-77 15:09 PAGE 1-1


38                ;SUBROUTINE TO SEND 2 PARTS OF ADDRESS(10BITS)
39        020000          MASK2=020000
40 00134 017500 ADDXMT:   MOV     @2(R5),R0       ;GET ADDERSS 0
          000002
41 00140 012702           MOV     #8.,R2          ;SHIFT COUNT
          000010
42 00144 010001           MOV     R0,R1
43 00146 006201 LOOP1:    ASR     R1
44 00150 005302           DEC     R2
45 00152 001375           BNE     LOOP1
46 00154 066701           ADD     MASK1,R1
          000036
47 00160 010137           MOV     R1,@#167772
          167772
48 00164 042700           BIC     #177400,R0
          177400
49 00170 062700           ADD     #MASK2,R0
          020000
50 00174 032737 LOOP2:    BIT     #400,@#167774
          000400
          167774
51 00202 001774           BEQ     LOOP2
52 00204 010037           MOV     R0,@#167772
          167772
53 00210 016500           MOV     4(R5),R0        ;GET ADDRESS OF DATA
          000004
54 00214 000204           RTS     R4
55 00216 000000 MASK1:     WORD   0

MBM SUBROUTINES MACRO V06-04A 07-JUL-77 15:09 PAGE 2


1         000001           END


MBM SUBROUTINES MACRO V06-04A 07-JUL-77 15:09 PAGE 2-1
SYMBOL TABLE

ADDXMT  000134R           LOOP1   000146R           LOOP2   000174R
LOOP3   000016R           LOOP4   000026R           LOOP5   000042R
LOOP6   000102R           LOOP7   000112R           MACRED  000064RG
MACWRT  000000RG          MASKR = 010400           MASKW = 011000
MASK1   000216R           MASK2 = 020000
  ABS.  000000      000
        000220      001
ERRORS DETECTED:  0
FREE CORE:  4712. WORDS
MBM1,MBM1<MBM1,END


*
```

```
 KB:<MBMV08.MAP


MAP OF MBMV08.LDA AT 09:13 ON 05-AUG-77 BY LINK  V03-03A

*** ROOT SEG: MAIN.

R/W MEM  LIMITS: 043036 077457 034422
IDENTIFICATION  : 02
PRG XFR ADDRESS: 043036


PROGRAM SECTION ALLOCATION SYNOPSIS.

<.  BLK.>: 043036 077457 034422
<.  ABS.>: 000000 000000 000000

*** TITLE: MAIN.                        FILE: MBMV08.OBJ

<.  BLK.>: 043036 062137 017102
    MAIN.   043036-R

*** TITLE: MBM                          FILE: MBM1  .OBJ

<.  BLK.>: 062140 062357 000220
    MACRED 062224-R  MACWRT 062140-R

MAX VIRTUAL ADDRESS 011053, V.M. BLOCKS USED 000000

#
```

APPENDIX VI

PROGRAM EXAMPLES

```
EXAMPLES OF ASCII WRITE AND READ ROUTINES
RU MBMV08


READ,WRITE,TEST OR DUMP
W

FILE NUMBER
10

WRITE IN DATA
12345678901234567890123456789012345678901234567890123456789012345678901234567890

WRITE IN DATA
12345678901234567890123456789012345678901234567890

WRITE IN DATA
12345678901234567890123456789012345678901234567890


READ,WRITE,TEST OR DUMP
R

FILE NUMBER
10

12345678901234567890123456789012345678901234567890
12345678901234567890123456789012345678901234567890
12345678901234567890123456789012345678901234567890
12345678901234567890123456789012345678901234

READ,WRITE,TEST OR DUMP
D

BEGIN FILE NUMBER?
10

END FILE NUMBER?
11


        ****************************************
MBM FILE DUMP
FILE NUMBER  10
12345678901234567890123456789012345678901234567890
12345678901234567890123456789012345678901234567890
12345678901234567890123456789012345678901234567890
12345678901234567890123456789012345678901234

        ****************************************
MBM FILE DUMP
FILE NUMBER  11
^P^P^P
```

```
EXAMPLE OF THE SPECIFY PATTERN TEST
S

WRONG ANSWER

READ, WRITE, TEST OR DUMP
T

TYPE OF TEST-FILE INTEGRITY, SPECIFY PATTERN OR GENERATE PATTERN?
S

BEGIN FILE NUMBER?
0

END FILE NUMBER?
25

NUMBER OF PATTERNS?
2

ENTER PATTERN NUMBER  1
0

ENTER PATTERN NUMBER  2
-1

MAX. NUMBER OF ERRORS TO BE PRINTED?
3

NUMBER OF WRITE PASSES?
1

NUMBER OF READS PER WRITE?
5


WRITE    READ     GOOD     BAD      BOARD    WORD
PASS     PASS     DATA     DATA
ACCUMULATED ERRORS=     0
END OF TEST ON FILES    0 THRU  25

READ, WRITE, TEST OR DUMP
T
TYPE OF TEST-FILE INTEGRITY, SPECIFY PATTERN OR GENERATE PATTERN?
S

BEGIN FILE NUMBER?
10
END FILE NUMBER?
11

NUMBER OF PATTERNS?
8

ENTER PATTERN NUMBER  1
1

ENTER PATTERN NUMBER  2
-3

ENTER PATTERN NUMBER  3
4

ENTER PATTERN NUMBER  4
-9
```

```
ENTER PATTERN NUMBER  5
16

ENTER PATTERN NUMBER  6
-33

ENTER PATTERN NUMBER  7
64

ENTER PATTERN NUMBER  8
-128

MAX. NUMBER OF ERRORS TO BE PRINTED?
10

NUMBER OF WRITE PASSES?
0

NUMBER OF READS PER WRITE?
1


WRITE   READ    GOOD    BAD     BOARD   WORD
PASS    PASS    DATA    DATA
FILE NUMBER  10
   1       1       1       0       0       0
   1       1      -3      -1       1       1
   1       1       4       0       2       2
   1       1      -9      -1       3       3
   1       1      16       0       4       4
   1       1     -33      -1       5       5
   1       1      64       0       6       6
   1       1    -128      -1       6       7
   1       1       1       0       0       8
   1       1      -3      -1       1       9
   1    ****MAX. NUMBER OF ERRORS( 10)****
FILE NUMBER  11
   1       1       1       0       0       0
   1       1      -3      -1       1       1
   1       1       4       0       2       2
   1       1      -9      -1       3       3
   1       1      16       0       4       4
   1       1     -33      -1       5       5
   1       1      64       0       6       6
   1       1    -128      -1       6       7
   1       1       1       0       0       8
   1       1      -3      -1       1       9
   1    ****MAX. NUMBER OF ERRORS( 10)****
ACCUMULATED ERRORS=   20
END OF TEST ON FILES  10 THRU  11

READ, WRITE, TEST OR DUMP
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

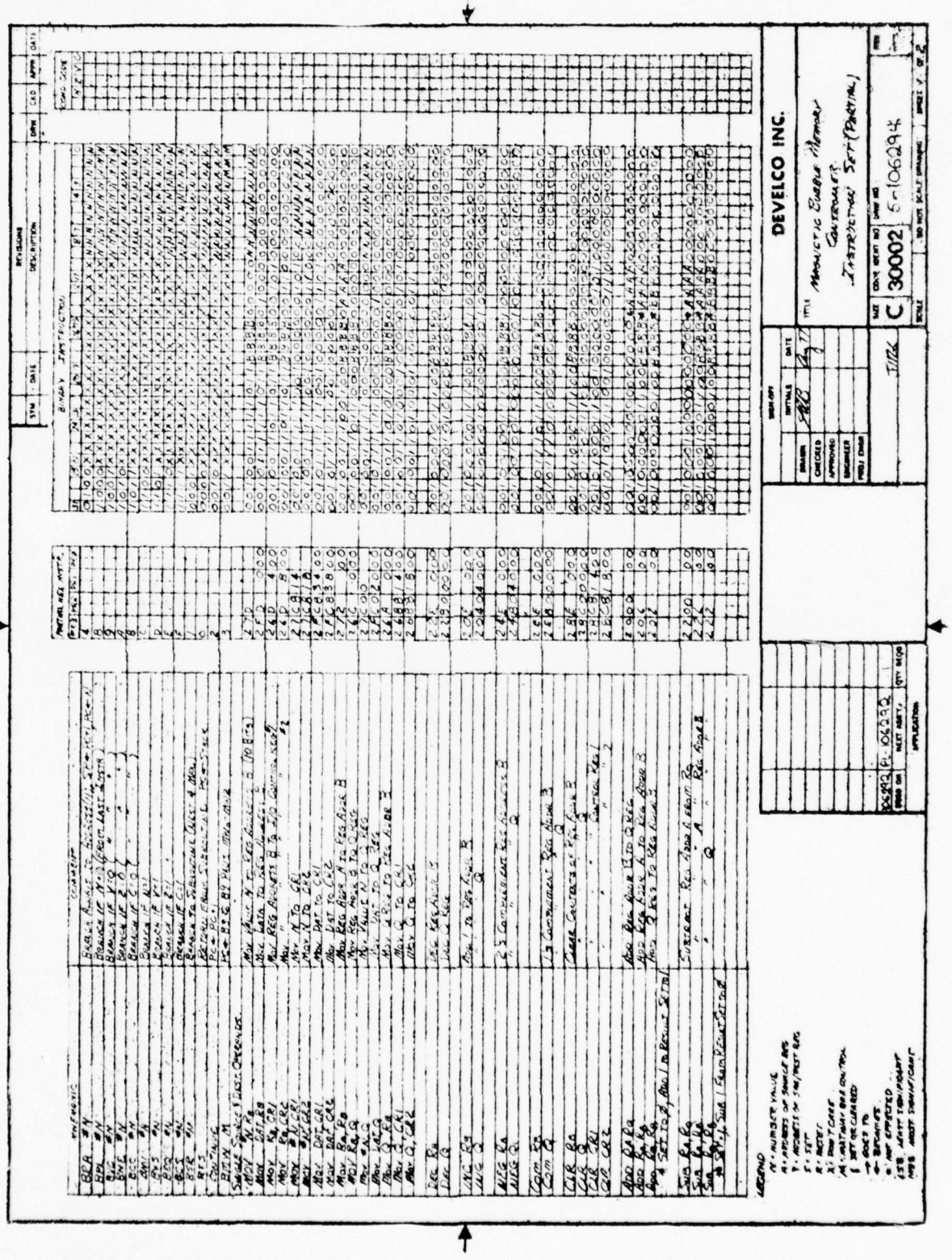
```
  EXAMPLE OF GENERATE PATTERN TEST

T

TYPE OF TEST-FILE INTEGRITY, SPECIFY PATTERN OR GENERATE PATTERN?
G

MAX. NUMBER OF ERRORS TO BE PRINTED?
3

NUMBER OF WRITE PASSES?
9

NUMBER OF READS PER WRITE?
1


WRITE   READ    GOOD    BAD     BOARD   WORD
PASS    PASS    DATA    DATA
PATTERN NUMBER 1
WORD0=   0, WORD1=  -1, ETC.
FILE NUMBER 488
   1       1       -1      -2      0       47
TOTAL ERRORS     1
ACCUMULATED ERRORS=    1
PATTERN NUMBER 2
WORD0=  -1, WORD1=   0, ETC.
   2    *****NO ERRORS****
PATTERN NUMBER 3
WORD0= -86, WORD1= -86, ETC.
   3    *****NO ERRORS****
PATTERN NUMBER 4
WORD0=  85, WORD1=  85, ETC.
   4    *****NO ERRORS****
PATTERN NUMBER 5
WORD0=   0, WORD1=   0, ETC.
   5    *****NO ERRORS****
PATTERN NUMBER 6
WORD0=  -1, WORD1=  -1, ETC.
   6    *****NO ERRORS****
MBM FILE INTEGRITY TEST

WRITE   READ    GOOD    BAD     BOARD   WORD
PASS    PASS    DATA    DATA
FILE NUMBER 281
   1       1       25      17      3       109
TOTAL ERRORS     1

WRITE   READ    GOOD    BAD     BOARD   WORD
PASS    PASS    DATA    DATA
   8       1    ***NO ERRORS***
ACCUMULATED ERRORS=    1

WRITE   READ    GOOD    BAD     BOARD   WORD
PASS    PASS    DATA    DATA
PATTERN NUMBER 1
WORD0=   0, WORD1=  -1, ETC.
   9    *****NO ERRORS****
TOTAL ACCUMULATED ERROR=    2
END OF GENERATE PATTERN TEST

READ, WRITE, TEST OR DUMP
```

```
EXAMPLE OF FILE INTEGRITY TEST

T

TYPE OF TEST-FILE INTEGRITY, SPECIFY PATTERN OR GENERATE PATTERN?
F

MAX. NUMBER OF ERRORS TO BE PRINTED?
3

NUMBER OF WRITE PASSES?
2

NUMBER OF READS PER WRITE?
2

MBM FILE INTEGRITY TEST

WRITE   READ    GOOD    BAD     BOARD   WORD
PASS    PASS    DATA    DATA
   1       1    ***NO ERRORS***
   1       2    ***NO ERRORS***

WRITE   READ    GOOD    BAD     BOARD   WORD
PASS    PASS    DATA    DATA
   2       1    ***NO ERRORS***
   2       2    ***NO ERRORS***
ACCUMULATED ERRORS=    0
END OF MBM FILE INTEGRITY TEST

READ, WRITE, TEST OR DUMP
```

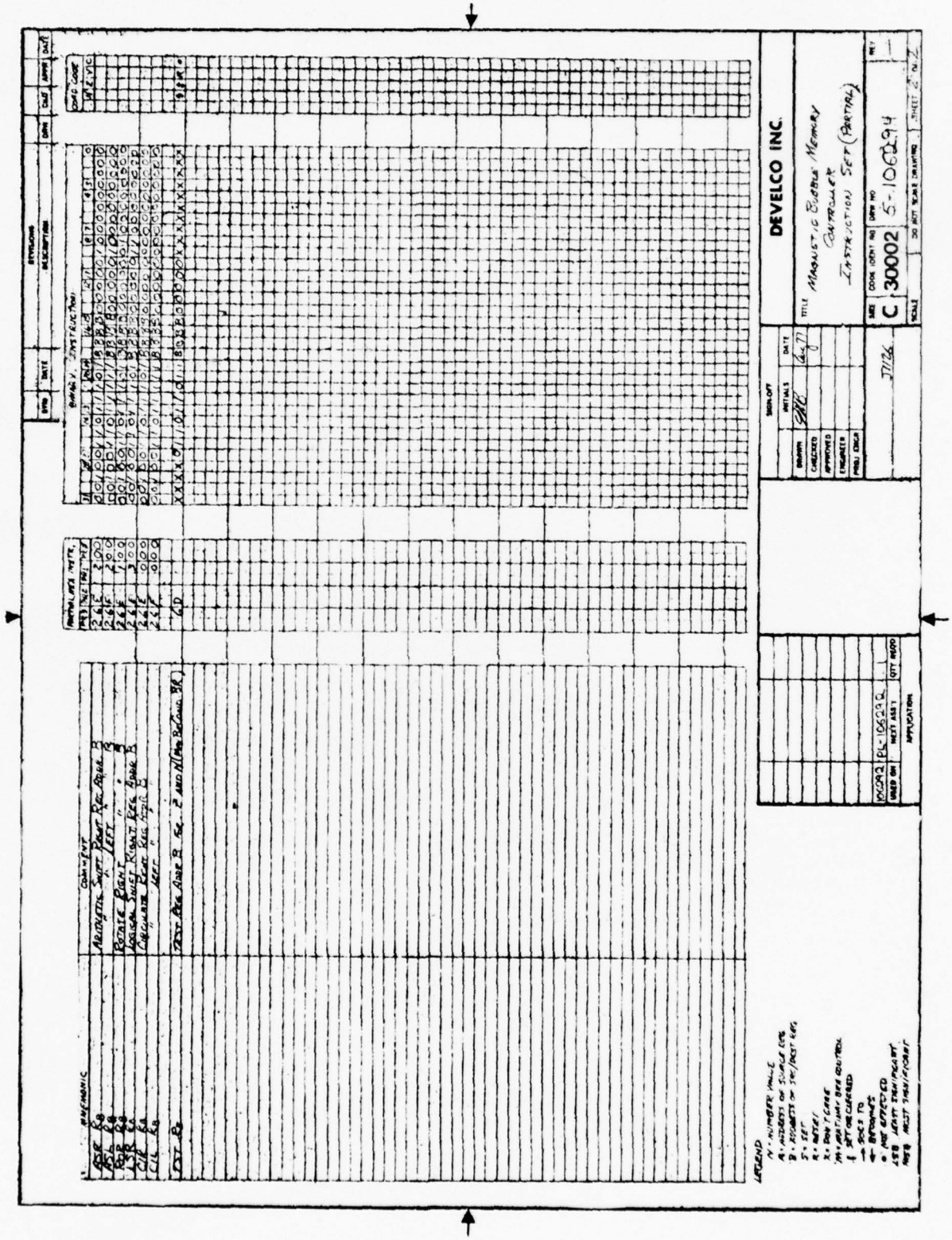
APPENDIX VII

MBM CONTROLLER INSTRUCTION SET (PARTIAL)

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC



THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC



APPENDIX VIII

MAGNETIC BUBBLE MEMORY PROGRAM

PRECEDING PAGE BLANK-NOT FILMED

SHEET #1 MAGNETIC BUBBLE MEMORY PROGRAMMING SHEET

LINE	ADDR HEX	INSTRUCTION HEX	BINARY EQUIVALENTS: BRANCH CONTROL	OP CODE	SRC CODE	DST CODE	B REG	C	A REG D ID	WAIT	ADDRESS DATA	EXTD CNTRL	INSTRUCTION
1	000	38 D8 14 00	0 0 1 1 1 0 0 0 1 1 0 1	0 0 0 0	0 0 1 0 1	0	0 0 0 0 0 0 0 0						CLR R0, BMWW
2	002	48 D8 98 08	0 1 1 0 0 1 0 0 0 1 1 0 1 1	0 0 0 1	0 0 1 1 0	0	0 0 0 0 1 0 0 0						CLR R1, BRA 008H (WRITE)
3													
4	004	48 D8 98 80	0 1 0 0 1 0 0 0 1 1 0 1 1	0 0 0 1	0 0 1 1 0	0	0 0 0 0 0 0 0 0						CLR R1, BRA 080H (READ)
5													
6													
7													
8	008	10 08 00 50	0 0 0 1 0 0 0 0 0 0 0 1	0 0 0 0	0 0 0 0 0	0	0 0 0 1 0 1 0 0						BSR 50H
9	009	27 D8 98 42	0 0 1 0 0 1 1 1 1 1 1 0	1 1 0 1	0 0 1 1 0	0	0 1 0 0 0 0 1 0						STO #42H, CRIERR, WRTEN, SOT ADDR
10	00A	27 D6 14 91	0 0 1 0 1 0 1 1 1 1 1 0	1 1 0 1	0 0 1 0 1	0	1 0 0 1 0 0 0 1						STO #91H, CRIERR, MBMEN, ALLEN, FIRQH
11	00B	27 D9 00 E4	0 0 1 0 1 0 1 1 1 1 1 0	1 1 0 1	0 0 1 0 0	0	0 0 1 1 0 0 1 0						STO #E4H, R2 ; SHFT CNT = 228.
12	00C	10 08 00 28	0 0 0 1 0 0 0 0 0 0 0 1	0 0 0 0	0 0 0 0 0	0	0 0 0 1 0 1 0 0						BSR 28H , WAIT FOR 0°
13	00D	22 D9 00 00	0 0 1 0 0 0 1 0 1 1 1 0	1 1 0 1	0 0 1 0 0	0	0 0 0 0 0 0 0 0						DEC R2
14	00E	A2 08 00 0C	1 0 1 0 0 0 1 0 0 0 0 0	0 0 0 1	0 0 0 0 0	0	0 0 0 1 1 0 0 0						BNE 00C
15	00F	27 D7 00 90	0 0 1 0 0 1 1 1 1 0 1 1	0 0 1 0	0 0 1 0 0	0	0 0 0 0 1 0 0 1						STO #90H, R2 , =157.
16	010	27 D8 14 99	0 0 1 0 1 0 1 1 1 1 0 1	1 0 0 0	0 0 1 0 0	0	1 0 0 1 1 0 0 1						STO #99H, CRIER0, GEN EN
17	011	10 08 00 28	0 0 0 1 0 0 0 0 0 0 0 1	0 0 0 0	0 0 0 0 0	0	0 0 0 1 0 1 0 0						BSR 28H , WAIT FOR 0°
18	012	27 D8 14 91	0 0 1 0 1 0 1 1 1 1 0 1	1 0 0 0	0 0 1 0 0	0	1 0 0 1 0 0 0 1						STO #91H, CRIER0 ; GEN INH.
19	013	10 08 00 28	0 0 0 1 0 0 0 0 0 0 0 1	0 0 0 0	0 0 0 0 0	0	0 0 0 1 0 1 0 0						BSR 28H ; WAIT FOR 0
20	014	22 D9 00 00	0 0 1 0 0 0 1 0 1 1 1 0	1 1 0 1	0 0 1 0 0	0	0 0 0 0 0 0 0 0						DEC R2
21	015	A2 08 00 11	1 0 1 0 0 0 1 0 0 0 0 1	0 0 0 0	0 0 0 0 0	0	0 0 0 1 0 0 0 1						BNE 10H
22	016	27 D7 00 1E	0 0 1 0 0 1 1 1 1 0 1 1	0 0 1 0	0 0 1 0 0	0	0 0 0 1 1 1 1 0						STO #1EH, R2 ; 30.
23	017	27 D8 A4 81	0 0 1 0 0 1 1 1 1 0 1 1	0 0 0 0	0 1 0 1 0	0	1 0 0 0 0 0 0 1						STO #91, CRIER0 ; ANY INH.
24	018	10 08 00 28	0 0 0 1 0 0 0 0 0 0 0 1	0 0 0 0	0 0 0 0 0	0	0 0 0 1 0 1 0 0						BSR 28H ; WAIT FOR 0°
25	019	22 D9 00 00	0 0 1 0 0 0 1 0 1 1 1 0	1 1 0 1	0 0 1 0 0	0	0 0 0 0 0 0 0 0						DEC R2
26	01A	A2 08 00 18	1 0 1 0 0 0 1 0 0 0 0 1	0 0 0 0	0 0 0 0 0	0	0 0 0 1 1 0 0 0						BNE 18H
27	01B	42 08 00 40	0 1 0 0 0 0 1 0 0 0 0 1	0 0 0 0	0 0 0 0 0	0	0 1 0 0 0 0 0 0						BRA 40H ; STO BUBBLES
28	C												
29	D												
30	E												
31	F												
32	020												
33	11												
34	21												
35	31												
36	41												
37	51												
38	61												
39	71	30 08 00 29	0 0 1 1 0 0 0 0 0 0 1 0	0 1 0 0	0 0 0 0 0	0	0 0 0 1 0 1 0 0						BMWI ; WAIT FOR 270°
40	81	40 08 00 30	0 1 0 0 0 0 0 0 0 0 1 0	0 0 0 0	0 0 0 0 0	0	0 0 0 1 1 0 0 0						BRA 30H ;
41	91												
42	A1												
43	B1												
44	C2												

MAGNETIC BUBBLE MEMORY PROGRAMMING SHEET
C5-106295
REV —

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

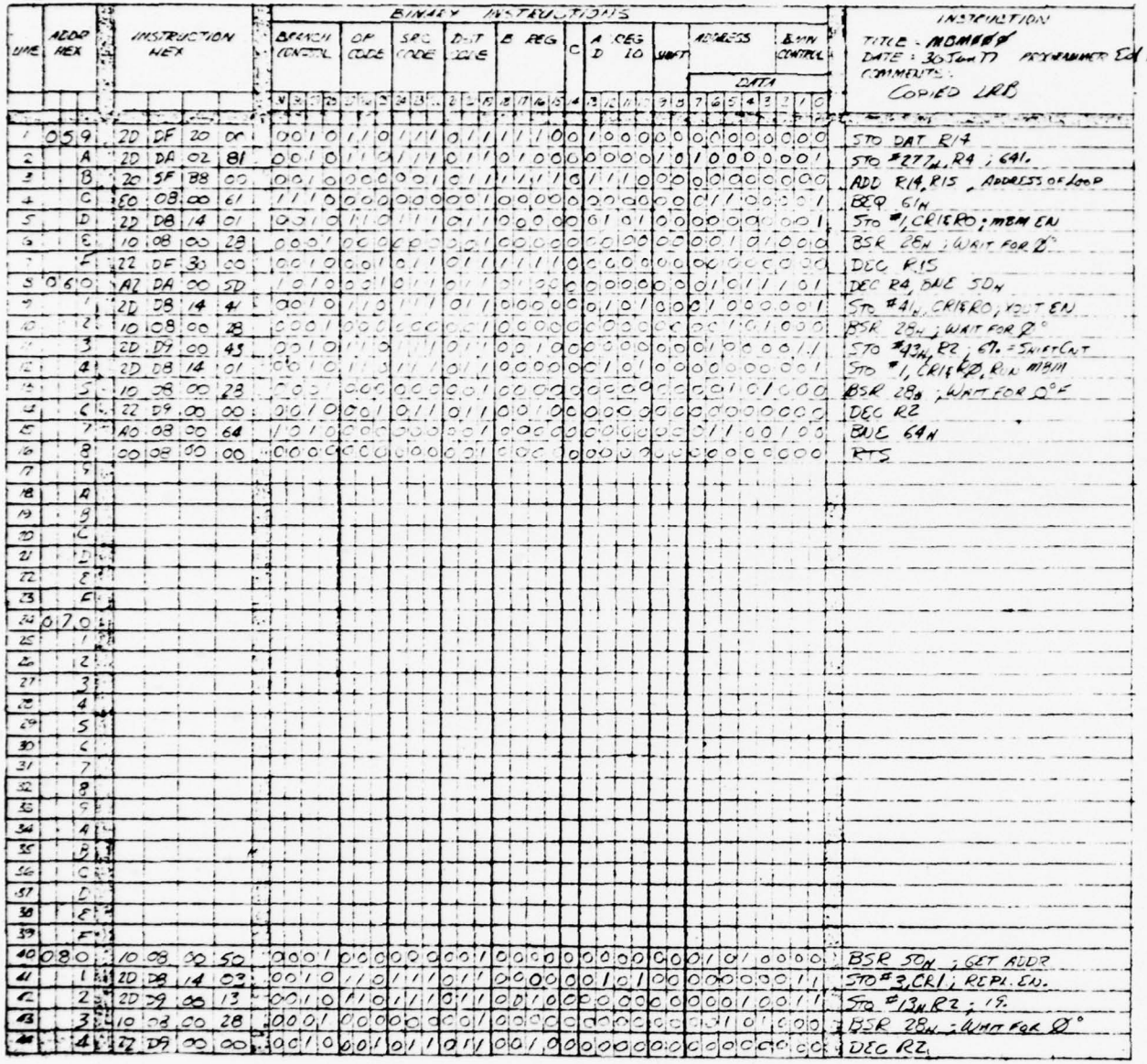
SHEET #2

MAGNETIC BUBBLE MEMORY PROGRAMMING SHEET

LINE	ADDRESS HEX	INSTRUCTION HEX	BINARY INSTRUCTIONS: BRANCH CONTROL	OP CODE	SRC CODE	DST CODE	R REG	C	A REG D IO	WAIT	ADDRESS / DATA	BYTE CONTROL	INSTRUCTION
1	02D												TITLE : MBM2000
2	E												DATE : 30 JAN 77 PROGRAMMER: CAH
3	F												COMMENTS:
4	030	30 08 00 32	0 0 1 1	0 0 0 0	0 0 0 0	0 1	0	0 0	0 0	0	0 0 1 1 0 0 1 0		COPIED LRB
5	031	00 08 00 00	0 0 0 0	0 0 0 0	0 0 0 0	0 0	0	0 0	0 0	0	0 0 0 0 0 0 0 0		BMW2 ; WAIT FOR 0°
6	2												RTS
7	3												
8	4												
9	5												
10	6												
11	7												
12	8												
13	9												
14	A												
15	B												
16	C												
17	D												
18	E												
19	F												
20	040	27 D8 14 A1	0 0 1 0	0 1 1 1	1 1 1 0	1 1	0	0 0	0 0	0	0 1 0 1 0 0 0 1		STO # A1H, CRI&RD, XIN EN
21	1	10 08 00 28	0 0 0 1	0 0 0 0	0 0 0 0	0 1	0	0 0	0 0	0	0 0 1 0 1 0 0 0		BSR 28H ; WAIT FOR 0°
22	2	27 D8 4 81	0 0 1 0	0 1 1 1	1 1 1 0	1 1	0	0 0	0 0	0	1 0 0 0 0 0 0 1		STO # 91, CRI&RD, XIN INH
23	3	22 DA 00 00	0 0 1 0	0 0 1 1	0 1 1 1	0 1	0	0 0	0 0	0	0 0 0 0 0 0 0 0		DEC R4 ; ORIG SET BY SBR #5@H
24	4	20 88 20 41	1 0 0 0	0 0 0 0	0 1 0 0	0 1	0	0 0	0 0	0	0 1 0 0 0 0 0 1		BPL 41H
25	5	20 08 00 00	0 0 1 0	0 0 0 0	0 0 0 0	0 1	0	0 0	0 0	0	0 0 0 0 0 0 0 0		NOP
26	6	20 D8 14 00	0 0 1 0	1 1 0 1	1 1 1 0	1 1	0	0 0	0 0	0	0 0 0 0 0 0 0 0		STO #0, CRI&RD, MBM INH
27	7	42 D8 98 00	0 1 0 0	1 1 0 1	1 1 1 0	1 1	0	0 0	0 1	0	0 0 0 0 0 0 0 0		STO #0, CRE&RD, EBRA
28	8												
29	9												
30	A												
31	B												
32	C												
33	D												
34	E												
35	04F												
36	050	30 08 00 54	0 0 1 1	0 0 0 0	0 0 0 0	0 1	0	0 0	0 0	0	0 1 0 1 0 1 0 0		BMW4 ; WAIT FOR UPPER ADDR
37	1	20 FF A2 00	0 0 1 0	1 1 0 0	1 1 1 1	1 1	0	0 1	0 0	1	0 0 0 0 0 0 0 0		ASL DAT 2 TO R15 ; BBSBB=ADDR
38	2	27 D9 00 27	0 0 1 0	0 1 1 1	1 1 1 0	1 1	0	0 1	0 0	0	0 0 0 0 0 0 1 1		STO #7, R2 ; SHIFT COUNT
39	3	27 D8 98 01	0 0 1 0	0 1 1 1	1 1 1 0	1 1	0	0 0	0 1	0	0 0 0 0 0 0 0 1		STO #1, CRI&RI, STB DATA RECEIVED
40	4	26 FF 82 00	0 0 1 0	0 1 1 0	1 1 1 1	1 1	0	0 0	0 0	0	0 0 0 0 0 0 0 0		ASL R15 ; SHIFT TO BBSB9
41	5	22 D9 00 00	0 0 1 0	0 0 1 0	0 1 1 1	0 1	0	0 0	0 0	0	0 0 0 0 0 0 0 0		DEC R2
42	6	40 08 00 53	1 0 1 0	0 0 0 0	0 0 0 0	0 1	0	0 0	0 0	0	0 1 0 1 0 0 1 1		BNE 53H
43	7	20 08 00 00	0 0 1 0	0 0 0 0	0 0 0 0	0 1	0	0 0	0 0	0	0 0 0 0 0 0 0 0		NOP
44	058	30 08 00 5D	0 0 1 1	0 0 0 0	0 0 0 0	0 1	0	0 0	0 0	0	0 1 0 1 1 1 0 1		BMW5 ; WAIT FOR REST OF ADDR

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

MAGNETIC BUBBLE MEMORY
PROGRAMMING SHEET
CS-105295
REV.—



SHEET #3

MAGNETIC BUBBLE MEMORY PROGRAMMING SHEET

LINE	ADDR HEX	INSTRUCTION HEX	BINARY INSTRUCTIONS: BRANCH CONTROL	OP CODE	SRC CODE	DST CODE	B REG	C	A REG D IO	WAIT	ADDRESS DATA	B/W CONTROL	INSTRUCTION
			N 3 2 1 0	5 4 3 2 1	4 3 2 1 0	4 3 2 1 0	4 3 2 1 0	4	3 2 1 0	3 2	7 6 5 4 3 2 1 0		TITLE - NOMAG# DATE : 30 JUN 77 PROGRAMMER: EAH COMMENTS: COPIED LRB
1	059	2D DF 20 00	0 0 1 0	1 1 0	1 1 1	0 1 1	1 1 1 1 0	0	1 0 0 0 0	0 0	0 0 0 0 0 0 0 0		STO DAT R14
2	A	2D DA 02 81	0 0 1 0	1 1 0	1 1 1	0 1 1	0 1 0 0 0	0	0 0 0 0 1	0 0	1 0 0 0 0 0 0 1		STO #277H, R4 ; 641.
3	B	20 3F B8 00	0 0 1 0	0 0 0	0 0 1	0 1 1	1 1 1 1 0	0	1 1 1 1 0	0 0	0 0 0 0 0 0 0 0		ADD R14, R15 , ADDRESS OF LOOP
4	C	60 08 00 61	1 1 1 0	0 0 0 0	0 0 0 0	0 0 1 0	0 0 0 0 0	0	0 0 0 0 0	0 0	0 1 1 0 0 0 0 1		BEQ 61H
5	D	2D D8 14 01	0 0 1 0	1 1 0	1 1 1	0 1 1	0 0 0 0 0	0	1 0 1 0 0	0 0	0 0 0 0 0 0 0 1		STO #1, CRIERO ; MEM EN
6	E	10 08 00 28	0 0 0 1	0 0 0 0	0 0 0 0	0 0 1 0	0 0 0 0 0	0	0 0 0 0 0	0 0	0 0 1 0 1 0 1 0		BSR 28H ; WAIT FOR Ø
7	F	22 DF 30 00	0 0 1 0	0 0 1	0 1 1	0 1 1	1 1 1 1 0	0	0 0 0 0 0	0 0	0 0 0 0 0 0 0 0		DEC R15
8	060	A2 DA 00 5D	1 0 1 0	0 0 1	0 1 1	0 1 1	0 1 0 0 0	0	0 0 0 0 0	0 0	0 1 0 1 1 1 0 1		DEC R4, BNE 5DH
9	1	2D D8 14 41	0 0 1 0	1 1 0	1 1 1	0 1 1	0 1 0 0 0	0	1 0 1 0 0	0 0	0 1 0 0 0 0 0 1		STO #41H, CRIERO, VOUT EN
10	2	10 08 00 28	0 0 0 1	0 0 0 0	0 0 0 0	0 0 1 0	0 0 0 0 0	0	0 0 0 0 0	0 0	0 0 1 0 1 0 0 0		BSR 28H ; WAIT FOR Ø°
11	3	2D D7 00 43	0 0 1 0	1 1 0	1 1 1	0 1 1	0 0 1 0 0	0	0 0 0 0 0	0 0	0 1 0 0 0 0 1 1		STO #43H, R2 ; 67. = SWEEP CNT
12	4	2D D8 14 01	0 0 1 0	1 1 0	1 1 1	0 1 1	0 0 0 0 0	0	1 0 1 0 0	0 0	0 0 0 0 0 0 0 1		STO #1, CRIERO, R14 MEM
13	5	10 08 00 28	0 0 0 1	0 0 0 0	0 0 0 0	0 0 1 0	0 0 0 0 0	0	0 0 0 0 0	0 0	0 0 1 0 1 0 0 0		BSR 28H ; WAIT FOR Ø°F
14	6	22 D9 00 00	0 0 1 0	0 0 1	0 1 1	0 1 1	0 1 0 0 0	0	0 0 0 0 0	0 0	0 0 0 0 0 0 0 0		DEC R2
15	7	A0 08 00 64	1 0 1 0	0 0 0	0 0 0	0 0 1	0 0 0 0 0	0	0 0 0 0 0	0 0	0 1 1 0 0 1 0 0		BNE 64H
16	8	00 02 00 00	0 0 0 0	0 0 0	0 0 0	0 0 1	0 0 0 0 0	0	0 0 0 0 0	0 0	0 0 0 0 0 0 0 0		RTS
17	9												
18	A												
19	B												
20	C												
21	D												
22	E												
23	F												
24	070												
25	1												
26	2												
27	3												
28	4												
29	5												
30	6												
31	7												
32	8												
33	9												
34	A												
35	B												
36	C												
37	D												
38	E												
39	F												
40	080	10 08 00 50	0 0 0 1	0 0 0 0 0	0 0 0 1 0	0 0 0 0 0	0 0 0 0 0	0	0 0 0 0 0	0 0	0 1 0 1 0 0 0 0		BSR 50H ; GET ADDR
41	1	2D D8 14 03	0 0 1 0	1 1 0	1 1 1	0 1 1	0 0 0 0 0	0	1 0 1 0 0	0 0	0 0 0 0 0 0 1 1		STO #3, CR1 ; REPL EN.
42	2	2D D9 00 13	0 0 1 0	1 1 0	1 1 1	0 1 1	0 0 1 0 0	0	0 0 0 0 0	0 0	0 0 0 1 0 0 1 1		STO #13H R2 ; 19.
43	3	10 08 00 28	0 0 0 1	0 0 0 0 0	0 0 0 1 0	0 0 0 0 0	0 0 0 0 0	0	0 0 0 0 0	0 0	0 1 0 1 0 0 0 0		BSR 28H ; WAIT FOR Ø°
44	4	D2 D9 00 00	1 0 0 1	0 0 1	0 1 1	0 1 1	0 0 1 0 0	0	0 0 0 0 0	0 0	0 0 0 0 0 0 0 0		DEC R2

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

MAGNETIC BUBBLE MEMORY
PROGRAMMING SHEET
C5-106295
REV. —

SHEET #4

MAGNETIC BUBBLE MEMORY PROGRAMMING SHEET

LINE	ADDR HEX	INSTRUCTION HEX				BINARY INSTRUCTIONS: 3 BRANCH CONTROL	OP CODE	SOURCE REG	DEST CODE	# REG	=	A REG OR IO	WAIT	ADDRESS / DATA	NUM CONTROL	INSTRUCTION
1	0B5	A5	D8	98	83	101	01111	011	0001	0	0	1	0	0	0	STO #82H CH2 RI, BNE 83H (RT3-D CH2 R2 CONT ADDR
2	6	2D	D9	00	90	001	11111	011	0010	0	0	0	0	1	0	STO #90H R2, 157.
3	7	2D	D8	14	87	001	11011	000	0001	0	1	0	0	1	1	STO #87H CH1 RT, REPL EN, STRB CN, FIFO EN
4	8	10	03	00	28	000	10000	000	0000	0	0	0	0	1	0	BSR 28H, WAIT FOR RT°
5	9	2D	D8	14	83	001	11011	000	0001	0	1	0	0	1	1	STO #83H CH1 RT, STROBE INH.
6	A	10	08	00	28	000	10000	000	0001	0	0	0	0	1	0	BSR 28H; WAIT FOR RT°
7	B	22	D9	00	00	001	00110	011	0010	0	0	0	0	0	0	DEC R2
8	C	40	08	00	E7	100	00000	000	0001	0	0	0	0	1	1	BNE E7H
9	D	2D	D9	00	EF	001	11011	011	0010	0	0	0	0	1	1	STO #EF, R2; 239.
10	E	22	D8	14	B1	001	11011	011	0000	0	1	0	0	1	1	STO #B1H CH1; REP INH
11	0BF	40	08	CC	18	010	00000	001	0000	0	0	0	0	0	0	BRA 18H

TITLE = MBM0000
DATE = 30JUN77 PROGRAMMER = D HUTCHINS
COMMENTS:
CODED BY LRB

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

MAGNETIC BUBBLE MEMORY
PROGRAMMING SHEET
C5-10629-5
REV.—

APPENDIX IX

EXAMPLE OF TEXAS INSTRUMENTS MBM TEST DATA

PRECEDING PAGE BLANK-NOT FILMED

FINAL MASK

MODULE 74-36-01 23MAR76 12:38:17 TEMP 1

0080 0200 0200 0000 0000 0200 000C 0000 0008 0007 NBIT 7

PULSE PARAMETERS

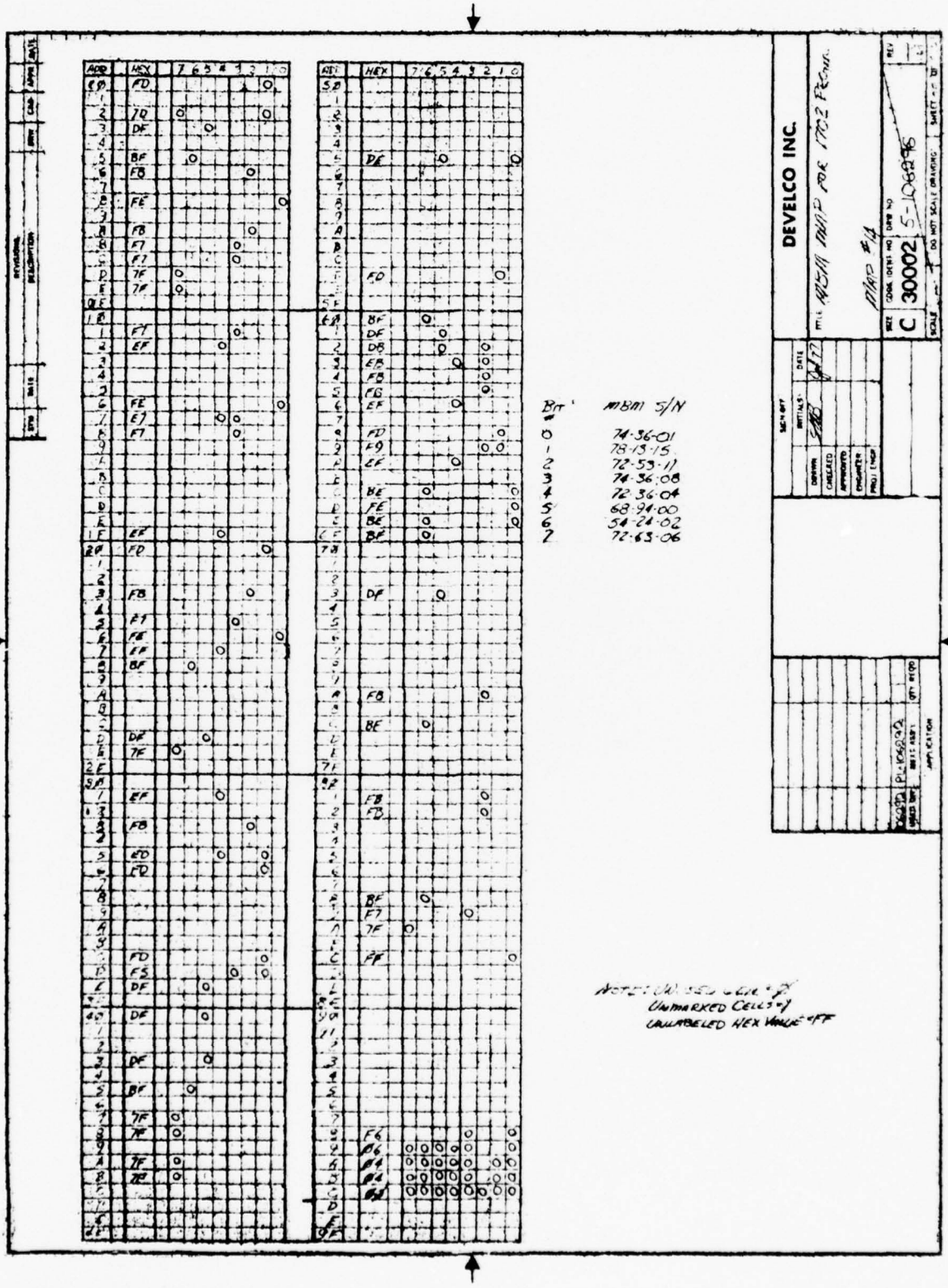
FUNCTION	LEADING EDGE	DURATION	AMPLITUDE
REPLICATE	5148 NS	1560 NS	132 MA
ANNIHILATE	3432 NS	4680 NS	60 MA
GENERATE	312 NS	468 NS	265 MA
TRANS OUT	1092 NS	1716 NS	43 MA
TRANS IN	4680 NS	2496 NS	30 MA

MODULE 74-36-01 23MAR76 12:38:20 TEMP 1

APPENDIX X

MBM MAP FOR 1702 PROM

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC



APPENDIX XI

PROM CODE FOR FUNCTION TIMING GENERATOR

74S471 PROM CODE FOR
FUNCTION TIMING GENERATOR

ADDRESS (MSB)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	08	00	08	00	08	00	08	00	08	00	08	00	08	00	08
10	00	08	00	08	00	08	00	08	00	08	00	08	00	08	00	08
20	00	08	00	08	00	08	00	08	00	08	04	08	04	08	04	08
30	04	08	04	08	04	08	04	08	04	08	04	08	04	08	14	08
40	14	08	10	08	10	08	10	08	10	08	10	08	10	08	10	08
50	10	08	18	09	18	08	08	08	88	08	88	0A	88	0A	88	0A
60	C8	0A	88	0A	88	0A	28	0A	28	08	20	08	20	08	21	08
70	21	08	21	08	21	08	21	0C	21	1C	25	1C	25	1C	05	1C
80	05	1C	05	1C	05	0C	05	0C	05	0C	05	08	05	08	15	08
90	15	08	11	08	10	08	10	08	10	08	10	08	10	08	10	28
A0	12	08	18	08	18	08	08	08	08	08	08	08	08	08	08	08
B0	08	08	08	08	08	08	28	08	28	08	20	08	20	08	20	08
C0	20	08	20	08	20	08	20	08	20	08	24	08	24	08	04	08
D0	04	08	04	08	04	08	04	08	04	08	04	08	04	08	04	08
E0	04	08	00	08	00	08	00	08	00	08	00	08	00	08	00	08
F0	00	08	00	08	00	08	00	08	00	08	00	08	00	08	00	00

ADDRESS (LSB)

PROM DATA IN HEXADECIMAL

APPENDIX XII

TEST REPORT MEMOS TO TEXAS INSTRUMENTS

RD7/1126-M286
16 August 1977

MEMO TO: Gerald Cox
Texas Instruments

cc: J1126
C.A. Snyder
T.C. Moore

FROM: Larry Bulduc, Develco

SUBJECT: Magnetic Bubble Memories

Under a separate cover, (9) TBM0101 memories - 8 of which are on boards - are being returned to you for reevaluation.

Our test consists of 8 patterns each of which is repeated every 8 write cycles. The pattern values consist of 1 or 2 8 bit words which are repeated in a file (8 bit wide system).

The patterns are as follows:

Pattern	Word Ø	Word 1	Reason
1	ØØØØØØØØ	11111111	Test for Alternate minor loop interference
2	11111111	ØØØØØØØØ	Complement of Pattern 1
3	10101010	10101010	System noise test
4	01010101	01010101	Complement of Pattern 3
5	00000000	00000000	To discriminate between system noise and effects of entire bubble in 1 state
6	11111111	11111111	Complement of Pattern 5

The above 6 patterns are written and verified a number of times into each file individually (e.g.; file numbers > the test file contain the last test pattern).

Patterns 7 and 8 are for testing file integrity.

Pattern 7 writes 4 bytes (to the 8 wide memory) which are repeated for the 144 bytes/file.

The four bytes are file address based

Word 0 - High byte of Address
Word 1 - Low byte of Address
Word 2 - 1's complement of Word Ø
Word 3 - 1's complement of Word 1

Pattern 8 is the complement of Pattern 7 (e.g.; Word Ø ← Word 2, Word 1 ← Word 3, Word 2 ← Word 0 and Word 3 ← Word 1).

In patterns 7 and 8 the entire memory is written first. Then each file is read 1 time in succession. The entire memory is read a specified number of times.

On August 5 - August 7 a test was conducted writing into the entire memory 610 times and reading the entire memory six times for every write. This would test each pattern approximately 78 times.

The number of bits read calculates to be

3.4×10^{8}/board
or 2.7×10^{9}/system.

A total of 112 errors were recorded on 5 of the 8 boards. This does not include successive errors of the same bit on the same write cycle but only the first occurrence.

Overall analysis of the test is shown in Table 1.

Repetitive errors occurred at the following word numbers (0-143)

TBM S/N	Word #	# Frequency
72-63-06	11	10
54-24-04	32	2
54-24-04	96	2
54-24-04	116	2
54-24-04	124	2
78-13-15	97	2
78-13-15	69	2

It is believed that the reoccurring Word errors on boards 4 and 5 are within normal probabilities for the number of failures. However it is also believed that Word 11 on Board 5 is bad.

Only 3 instances of file reoccurrences were found, each having a frequency of 2.

TBM S/N	Fib #	Word #
78-13-15	0	40 & 24
72-63-06	295	11
72-63-06	369	11

However the frequency of errors did tend to increase with the file number as shown in Table 2.

Our current measurements on these boards are shown in Table 3. x and y coil currents were measured at 12.5 V, all other currents are at 12.0 volts.

In order to insure the above results were not our system problems, the (4) lowest error rate boards were exchanged with the highest error rate boards.

The test was repeated for 100 write cycles. All errors moved with the bubbles and bubble boards. No errors were detected in positions previously in error. It was therefore concluded that the cause of errors were not related to components or wiring in our system which are unique to each bit position.

If you have any questions please call me at (408) 734-5700.

Again I would like to repeat that your expedient attention to this matter is of the utmost importance to us in completing our report, in that we wish the report to be as complete and accurate as possible.

Larry Bulduc
Project Engineer

LB:nlh

TABLE 1

TBM S/N	BRD S/N	Bit Position	Failures vs. Pattern								Total Failures	Types of Failures			Hard Errors on 1st Read	Bubble	
			1	2	3	4	5	6	7	8		Soft	Hard	Error Shift		Lost	Gained
72-53-11	09	0	5	1	1	0	2	0	1	3	13	2	10	1	6	4	9
74-36-08	05	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74-36-01	01	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
78-13-15	08	3	9	9	0	0	0	0	5	11	34	0	33	1	33	1	33
54-24-02	02	4	7	13	1	0	0	0	7	15	43	0	40	3	38	3	40
72-63-06	07	5	4	4	5	0	1	2	0	4	20	0	19	1	12	2	18
72-63-04	06	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
68-94-00	03	7	0	1	0	0	1	0	0	0	2	0	1	1	1	1	1
		TOTAL	28	28	7	0	4	2	13	33	112	2	103	7			

TABLE 2

TBM S/N →	72-53-11	78-13-15	54-24-02	72-63-06	68-94-00	TOTAL
File #						
0-49	0	4	4	1	1	10
50-99	0	0	2	1	0	3
100-149	0	5	0	2	0	7
150-199	1	2	1	0	0	4
200-249	3	5	2	1	0	11
250-299	0	3	4	2	0	9
300-349	1	2	4	1	0	8
350-399	1	1	2	4	0	8
400-449	0	0	4	2	0	6
450-499	1	1	5	0	1	8
500-549	2	4	3	4	0	13
550-599	3	5	6	1	0	15
600-649	1	3	5	1	0	10

TABLE 3

CURRENT (mA)

MBM S/N	I_X	I_Y	I_{XO}	I_{XO}	I_R	I_A	I_G
72-63-06	±470	±480	39	28	120	55	240
74-36-08	±460	±475	39	28	116	55	245
72-53-11	±465	±480	39	25	120	56	230
54-24-02	±455	±475	38	29	120	55	230
78-13-15	±460	±485	40	30	122	57	235
68-94-00	±455	±480	40	30	124	58	230
74-36-01	±465	±475	40	31	125	55	235
72-63-04	±465	±485	40	25	120	55	240

RD7/1126-M276
5 August 1977

MEMO TO: Gerald Cox
Texas Instruments

cc: J1126 File

FROM: Larry Bulduc, Develco

SUBJECT: Magnetic Bubble Memories

Presently being shipped to T.I. are (5) TBM0103 memories which I believe are defective.

Our findings on each are as follows:

S/N 74-19-12

Analysis - Minor Loop #114 defective in addition to original 13 bad loops.

Test Conditions (mA)

$I_X = \pm 470$

$I_Y = \pm 485$

$I_G = 230$

$I_{XO} = 41$

$I_{XI} = 29$

$I_R = 130$

$I_A = 55$

Comments - This mode of failure in the loop would not maintain a bubble.

S/N 74-24-07

Analysis - When writing all bubbles into a file exhibits soft errors at a rate of $60/.9 \times 16^6$. This unit previously exhibited 0 errors in 8.2×10^7 bits when writing 2 bytes of the file address and complement in each file.

Example: Odd write passes ADD_H

ADD_L

$\overline{ADD_H}$

$\overline{ADD_L}$

Even write passes $\overline{ADD_H}$

$\overline{ADD_L}$

ADD_H

ADD_L

Each write was verified 6 times. In failure mode all bubbles were annihilated, then each file was written, read 10 times before proceeding to the next file. Another unit was installed in the same board and did not exhibit this problem.

Test Conditions

I_X = 480	I_{XI} = 30
I_Y = 490	I_R = 120
I_G = 230	I_A = 55
I_{XO} = 39	

Note: I_R pulse width = 5.25 μs.

S/N 74-20-02

Analysis - Minor loops 107, 115, 116, 117, 127 defective plus "random" errors.

Test Conditions

I_X = 460	I_{XI} = 29
I_Y = 480	I_R = 120
I_G = 220	I_A = 53
I_{XO} = 40	

Comments - Collapsing field produced no improvement. Error rate >1/92,000.

S/N 74-20-10

Analysis - Random errors. Error rate approx. $25/5x10^5$ before collapsing field. Error rate = $12/5x10^5$ after collapsing field. Reduced error rate obtained by increasing X, Y coil drive. At 5% over maximum spec error rate $\approx 3/5x10^5$.

Test Conditions

Not known precisely but within spec limits for 25°C.

S/N 72-63-15

Analysis - loops 124, 140 defective. Originally 12 loops defective. Collapsing field reduced error rate from $4/5x10^5$ to $5/2.5x10^6$.

Test Condition

Not known precisely but within specification limits for 25°C.

For your further information:

The bad loops in thirteen of the fourteen TBM0103's in our possession had been written into. This was the result of an earlier discussion with TI which at that time lead us to believe that it was allowable to write into bad loops. (The one exception is S/N 74-19-12 which is being returned to you.)

As a result I attempted collapsing the fields in the TBM0103's that were producing errors. The procedure was per your instructions in our telephone conversation of 28 July 77.

E.G. using a magnet with a field strength of 40 Oe or greater, introduce the field perpendicular to the front surface of the 0103 at a distance that erases the bubbles.

This procedure worked on 3 of 7 units. The remaining 7 units were apparently not affected by writing in the bad loops as they have error rates of $<1/2.6x10^7$.

Digital timing on our system was measured as follows:

All leading edges with respect to $\overline{CXB}$

SIGNAL	LEADING EDGE (μs)	WIDTH (μs)
CXB	0	3.0
CYB	2.5	3.0
CXA	5.0	3.0
CYA	7.5	3.0

SIGNAL	LEADING EDGE (μs)	WIDTH (μs)
Generator	.125	.25
Replicate	4.75*	1.50
Annihilate	3.50	4.75
X_{out}	1.125	1.75
X_{in}	4.75	2.5
Clamp	.75	1.75
Strobe	1.75	.25

*Actual test conditions produced by original timing map supplied by TI. Changed to 5.25 μs after these tests.

Your prompt attention to these units will enable us to accurately complete our report to our customer.

Original Develco P.O. #9786

Returned under TI RMR #DL9190-10

Larry Bulduc
Project Engineer

LB:lc

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

Director
Defense Advanced Rsch. Proj. Agency
ATTN: Tech. Lib.
ATTN: NMRO
ATTN: PMO
ATTN: STO

Defense Documentation Center
Cameron Station
12 cy ATTN: TC

Director
Defense Nuclear Agency
ATTN: DDST
ATTN: TISI Archives
2 cy ATTN: SPSS
3 cy ATTN: TITL, Tech. Lib.

Commander, Field Command
Defense Nuclear Agency
ATTN: FCTMOF
ATTN: FCT
ATTN: FCPR

Chief
Livermore Division, Field Command, DNA
Lawrence Livermore Laboratory
ATTN: FCPRL

Commandant
NATO School (SHAPE)
ATTN: U.S. Documents Officer

Under Secretary of Def. for Rsch. & Engrg.
ATTN: AE
ATTN: S&SS(OS)

DEPARTMENT OF THE ARMY

Dep. Chief of Staff for Rsch. Dev. & Acq.
ATTN: Tech. Lib.

Commander
Harry Diamond Laboratories
ATTN: DELHD-NP
ATTN: DELHD-TI, Tech. Lib.

Director
U.S. Army Ballistic Research Labs.
ATTN: DRXBR-X, Julius J. Meszaros
ATTN: DRDAR-BLD, J. H. Keefer
ATTN: Tech. Lib., Edward Baicy

Director
U.S. Army Engr. Waterways Exper. Sta.
ATTN: Tech. Lib.
ATTN: William Flathau
ATTN: F. Hanes
ATTN: J. K. Ingram

Commander
U.S. Army Materiel Dev. & Readiness Cmd.
ATTN: Tech. Lib.

DEPARTMENT OF THE NAVY

Chief of Naval Research
ATTN: Tech. Lib.

Officer-in-Charge
Civil Engineering Laboratory
Naval Construction Battalion Center
ATTN: R. J. Odello
ATTN: Tech. Lib.

Commander
David W. Taylor Naval Ship R&D Ctr.
ATTN: Code L42-3, Library

Commander
Naval Facilities Engineering Command
Headquarters
ATTN: Tech. Lib.

Commander
Naval Ship Engineering Center
ATTN: Tech. Lib.

Officer-in-Charge
Naval Surface Weapons Center
ATTN: Code WA501, Navy Nuc. Prgms. Off.
ATTN: Tech. Lib.

DEPARTMENT OF THE AIR FORCE

AF Geophysics Laboratory, AFSC
ATTN: SUOL, Rsch. Lib.

AF Institute of Technology, AU
ATTN: Library AFIT, Bldg. 640, Area B

AF Weapons Laboratory, AFSC
ATTN: DEX, J. Renick
ATTN: SUL
ATTN: DED-I, Capt Ray

Hq. USAF/IN
ATTN: INATA

DEPARTMENT OF ENERGY

Department of Energy
Albuquerque Operations Office
ATTN: Doc. Con. for Tech. Lib.

Department of Energy
Division of Headquarters Services
ATTN: Doc. Con. for Class. Tech. Lib.

Department of Energy
Nevada Operations Office
ATTN: Doc. Con. for Tech. Lib.

University of California
Lawrence Livermore Laboratory
ATTN: Tech. Info., Dept. L-3

Sandia Laboratories
Livermore Laboratory
ATTN: Doc. Con. for Tech. Lib.

DEPARTMENT OF ENERGY (Continued)

Sandia Laboratories
ATTN: Doc. Con. for Luke J. Vortman
ATTN: Doc. Con. for Org. 3422-1, Sandia Rpt. Coll.
ATTN: Doc. Con. for A. J. Chaban

DEPARTMENT OF DEFENSE CONTRACTORS

Aerospace Corporation
ATTN: Tech. Info. Services
ATTN: Prem N. Mathur

Agbabian Associates
ATTN: M. Agbabian

ARTEC Associates, Inc.
ATTN: D. W. Baum

The Boeing Company
2 cy ATTN: R. Holmes

Civil/Nuclear Systems Corp.
ATTN: Robert Crawford

DEVELCO, Inc.
ATTN: L. R. Buldue
ATTN: O. L. Kwong
ATTN: E. P. Hutchins
2 cy ATTN: T. C. Moore
2 cy ATTN: C. A. Snyder

EG&G Washington Analytical Services Center, Inc.
ATTN: Tech. Lib.

Electromechanical Sys. of New Mexico, Inc.
ATTN: R. A. Shunk

General Electric Company
TEMPO-Center for Advanced Studies
ATTN: DASIAC

H-Tech Laboratories, Inc.
ATTN: B. Hartenbaum

IIT Research Institute
ATTN: Tech. Lib.

Kaman Sciences Corporation
ATTN: Library
ATTN: Donald C. Sachs
ATTN: Eldine Cole

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Merritt CASES, Inc.
ATTN: J. L. Merritt
ATTN: Tech. Lib.

Nathan M. Newmark
Consulting Engineering Services
ATTN: W. Hall

Physics International Company
ATTN: Doc. Con. for Tech. Lib.
ATTN: Doc. Con. for Fred M. Sauer/
Charles Godfrey
ATTN: Doc. Con. for Coye Vincent

R&D Associates
ATTN: J. G. Lewis
ATTN: Tech. Lib.
ATTN: W. Graham

The Mitre Corporation
ATTN: Library

Science Applications, Inc.
ATTN: Tech. Lib.

Southwest Research Institute
ATTN: Wilfred E. Baker
ATTN: A. B. Wenzel

SRI International
ATTN: Burt R. Gasten/P. De Carli
ATTN: George R. Abrahamson

Systems, Science & Software, Inc.
ATTN: Tech. Lib.
ATTN: Donald R. Grine

TRW Defense & Space Sys. Group
ATTN: Peter K. Dai, R1/2170
ATTN: Tech. Info. Center/S-1930
ATTN: Paul Lieberman

Texas Instruments
ATTN: Mr. Gerald Cox

TRW Defense & Space Sys. Group
San Bernardino Operations
ATTN: E. Y. Wong, 527/712

The Eric H. Wang
Civil Engineering Rsch. Fac.
ATTN: Neal Baum